

Пояснювальна записка  
до дипломного проекту

---

на тему: Система аналізу та обробки даних на базі CRM

Київ - 2019

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ .....	4
ВСТУП .....	7
1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ.....	10
1.1 Поняття CRM системи, можливості .....	10
1.1.1 Класифікація CRM-систем .....	13
1.1.2 Операційні CRM .....	13
1.1.3 Аналітичні CRM .....	14
1.1.3 Колабораційні CRM.....	14
1.1.4 Комбіновані CRM-системи .....	15
1.2 Salesforce CRM .....	15
1.3 HubSpot CRM.....	18
1.4 Microsoft Dynamics CRM.....	20
1.5 Bitrix24 .....	22
1.6 Висновки дослідження існуючих CRM систем .....	23
2. ДОСЛІДЖЕННЯ ПРОБЛЕМИ ПРОГНОЗУВАННЯ.....	25
2.1 Загальні відомості .....	25
2.2 Причини виникнення проблеми прогнозування.....	29
2.3 Сучасні тенденції .....	30
2.3.1 Автоматизація і коммодування процесів обробки даних.....	32
2.3.2 Розвиток аналітичних систем реального часу.....	32
2.3.3 Поява нових інструментів для аналізу даних датчиків і різних пристроїв .....	33
2.3.4 Розвиток когнітивних систем і систем автоматичного прийняття рішення .....	33
2.3.5 Зміни в законодавстві, пов'язані з зберіганням і обробкою даних .....	34
2.4 Висновки .....	34

Зм	Арк.	№ документа	Підпис	Дата

ІАз51.070БАК.001 ПЗ

Аркуш

2

3. РОЗРОБКА І ОПИС ДОДАТКУ АНАЛІЗУ ДАНИХ.....	35
3.1 Опис алгоритму програми.....	35
3.2 Опис алгоритму аналізу даних Random Forest.....	36
3.2 Опис користувацької частини.....	43
3.3 Опис технології розробки .....	47
3.3.1 Реалізація Random Forest мовою програмування R. ....	47
3.3.2 Засоби інтеграції Salesforce CRM з алгоритмом Випадкових лісів. ....	51
3.3.3 Засоби реалізації користувацького інтерфейсу. ....	53
<b>ВИСНОВКИ .....</b>	<b>55</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>	<b>57</b>
<b>ДОДАТОК А.....</b>	<b>60</b>
<b>ДОДАТОК Б .....</b>	<b>61</b>
<b>ДОДАТОК В .....</b>	<b>62</b>
<b>ДОДАТОК Г .....</b>	<b>64</b>
<b>ДОДАТОК Д.....</b>	<b>65</b>
<b>ДОДАТОК Е .....</b>	<b>66</b>
<b>ДОДАТОК Ж.....</b>	<b>67</b>
<b>ДОДАТОК З.....</b>	<b>69</b>

## ПЕРЕЛІК СКОРОЧЕНЬ

CRM – Customer Relationship Management

Інтерфейс – набір інструментів для взаємодії користувача та комп'ютерної системи; візуальна частина, з якою взаємодіє користувач.

Фреймворк – готовий до використання комплекс програмних рішень, що охоплює дизайн, логіку та базову функціональність системи чи підсистеми.

API (прикладний програмний інтерфейс) – набір готових класів, процедур, функцій, структур і констант, що надаються застосунком (бібліотекою, сервісом) чи операційною системою для використання у зовнішніх програмних продуктах.

CLI (Command Line Interface) – інтерфейс командного рядка, консоль.

HTML (HyperText Markup Language) – стандартна мова розмітки веб-сторінок у мережі Інтернет.

HTTP – протокол передачі даних, що використовується у інфокомунікаційних мережах.

IDE (Integrated Development Environment) – інтегроване середовище розробки.

JavaScript (JS) – мова програмування високого рівня.

ETL — Extract, Transform, and Load (ETL) або Витяг, Перетворення та Завантаження — процес, який використовується в базах даних та, особливо, у сховищах даних та у засобах Business Intelligence для забезпечення їх роботи для підтримки прийняття рішень.

					<i>ІА351.070БАК.001 ПЗ</i>	Аркуш
						4
Зм	Арк.	№ документа	Підпис	Дата		

SaaS — Програ́ма як по́слуга (англ. Software as a service, SaaS) — модель поширення програм споживачам, при якій постачальник розробляє веб-програму, розміщує її й управляє нею (самостійно або через третіх осіб) з метою використання її замовниками через інтернет. Замовники платять не за володіння програмами як такими, а за їх використання (через API, що доступне через веб і яке часто використовують веб-служби). Близьким до терміну SaaS є термін «On-Demand» (за запитом).

PaaS - Платфо́рма як по́слуга (англ. Platform as a service, PaaS) — модель надання хмарних обчислень, при якій споживач отримує доступ до використання інформаційно-технологічних платформ: операційних систем, систем управління базами даних, зв'язного програмного забезпечення, засобів розробки і тестування розміщених у хмарних провайдерах. У цій моделі вся інформаційно-технологічна інфраструктура, включаючи обчислювальні мережі, сервери, системи зберігання, цілком керується провайдером, ним же визначається набір доступних для споживачів видів платформ та набір керованих параметрів платформ, а споживачеві надається можливість використовувати платформи, створювати їх віртуальні екземпляри, встановлювати, розробляти, тестувати, експлуатувати на них прикладне програмне забезпечення, при цьому динамічно змінюючи кількість споживаних обчислювальних ресурсів.

Force.com — програмна платформа Salesforce.com, на якій розроблені Sales Cloud і Service Cloud, надавана передплатникам для самостійної розробки додатків і розширень для CRM-системи Salesforce.com.

Apex - Java-подібна мова, розроблена компанією Salesforce.com для самостійної розробки додатків і розширень для CRM-системи Salesforce.com.

					<i>IA351.070БАК.001 ПЗ</i>	Аркуш
						5
Зм	Арк.	№ документа	Підпис	Дата		

Visualforce, Lightning Component фреймворк – засоби створення власних сторінок та додатків з вихідним форматом на основі XML, що забезпечує генерацію користувальницьких HTML / AJAX- і Flex-інтерфейсів, розроблений компанією Salesforce.com для самостійної розробки додатків і розширень для CRM-системи Salesforce.com.

Python («Пайтон») — інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується кілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована.

R — мова програмування і програмне середовище для статистичних обчислень, аналізу та зображення даних в графічному вигляді. R має значні можливості для здійснення статистичних аналізів, включаючи лінійну і нелінійну регресію, класичні статистичні тести, аналіз часових рядів (серій), кластерний аналіз і багато іншого. R легко розбудовується завдяки використанню додаткових функцій і пакетів доступних на сайті Comprehensive R Archive Network (CRAN). Більша частина стандартних функцій R, написана мовою R, однак існує можливість підключати код написаний C, C++, або Фортраном. Також за допомогою програмного коду на C або Java [5] можна безпосередньо маніпулювати R об'єктами.

					<i>IA351.070БАК.001 ПЗ</i>	Аркуш
						6
Зм	Арк.	№ документа	Підпис	Дата		

## ВСТУП

У зв'язку з удосконаленням технологій запису та зберігання даних людство отримало колосальні потоки інформаційної “руди” в усіляких областях. Розвиток методів запису і зберігання даних привів до бурхливого зростання обсягів збираної та аналізованої інформації. Обсяги даних настільки великі, що людина просто не в змозі проаналізувати їх самотійно, хоча необхідність проведення такого аналізу цілком очевидна, адже в цих “сирих даних” є знання, які можуть бути використані для прийняття рішень.

Діяльність будь-якого підприємства (комерційного, виробничого, медичного, наукового тощо) тепер супроводжується реєстрацією та записом усіх подробиць його діяльності. Став очевидним той факт, що без продуктивної переробки, потоки сирих даних нікому не потрібні. Специфіка сучасних вимог до такої переробки є такою:

1. дані мають необмежений обсяг;
2. дані є різнорідними (кількісними, якісними, текстовими);
3. результати повинні бути конкретні та зрозумілі;
4. інструменти для обробки сирих даних повинні бути прості у використанні.

Інструментів, призначених аналізувати накоплені дані, досить багато, але більшість з них є досить дорогими, тож малі та середні підприємства часто не можуть їх собі дозволити. Крім цього, такі інструменти складні у використанні, адже вимагають підключення до баз даних, що викликає труднощі, особливо якщо компанія не має власного ІТ департаменту, спеціалісти якого могли б займатися цією задачею.

					ІА351.070БАК.001 ПЗ	Аркуш
						7
Зм	Арк.	№ документа	Підпис	Дата		

Предметна область досліджень – підприємницька діяльність, об’єкт дослідження – засоби підвищення продуктивності підприємства та рівню продаж, предмет дослідження – система на основі CRM, призначена для аналізу даних угод та прогнозування вірогідності їх успіху.

Мета цього дипломного проекту – спроектувати гнучку систему, для використання будь-якими підприємствами, незалежно від структури, обсягів даних та бізнес-процесів, а також вона повинна мати можливість обробляти будь-які обсяги різномірної інформації щодо угод компанії з метою прогнозування вірогідності успіху кожної з угод.

Програмний продукт також має бути доступним усім підприємствам незалежно від рівня прибутку.

Система має бути легкою у підключенні, налагодженні та використанні, отже вона має бути реалізована як додаток до CRM-системи, в якій компанія зберігає усі дані об клієнтах, угодах, можливостях, контактах, продуктах, ін. Інтерфейс додатку має бути сучасним, простим та інтуїтивно-зрозумілим.

В результаті, потрібно отримати додаток, призначений для використання не тільки бізнес-аналітиками, але будь-якими співробітниками компанії. Усі працівники, що мають відношення до продаж, матимуть змогу користуватися додатком в повсякденній роботі з клієнтами та їх даними, а також використовувати результати прогнозування для прийняття оперативних рішень щодо об’єктів, які знаходяться під найбільшим ризиком, та планування підприємницької діяльності.

Потенційно, ця система даватиме можливість компаніям підняти ефективність праці та рівень прибутку.

В першому розділі необхідно проаналізувати найбільш розповсюджені CRM-системи, на основі яких можливо розробити

					ІА351.070БАК.001 ПЗ	Аркуш
						8
Зм	Арк.	№ документа	Підпис	Дата		



додаток, їх переваги, недоліки та доцільність їх використання в системі, також зробити висновки щодо вибору конкретної системи, обґрунтувати цей вибір.

В другому розділі необхідно дослідити проблему прогнозування в сучасному підприємницькому секторі та інших сферах людської діяльності, обґрунтувати актуальність цієї проблеми та сформулювати вимоги до продукту, що має вирішувати цю проблему.

В третьому розділі необхідно реалізувати програмний додаток до CRM-системи, що задовольнятиме поставленим вимогам, зобразити структуру системи в цілому, описати алгоритм роботи системи з додатком, алгоритм роботи власне додатку, та алгоритм, застосований для аналізу даних та прогнозування. Також необхідно описати користувацький інтерфейс та технології, використані для розробки додатку.

					ІА351.070БАК.001 ПЗ	Аркуш
						9
Зм	Арк.	№ документа	Підпис	Дата		

# 1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

## 1.1 Поняття CRM системи, можливості

Управління відносинами з клієнтами (англ. Customer relationship management (CRM), укр. сі-ар-ем) — поняття, що охоплює концепції, котрі використовуються компаніями для управління взаємовідносинами зі споживачами, включаючи збір, зберігання й аналіз інформації про споживачів, постачальників, партнерів та інформації про взаємовідносини з ними.

Сучасна CRM направлена на вивчення ринку і конкретних потреб клієнтів. На основі цих знань розробляються нові товари або послуги і таким чином компанія досягає поставлених цілей і покращує свій фінансовий показник.

Існує три *CRM-підходи*, кожен з яких може бути реалізованим окремо від інших:

1. Оперативний — автоматизація споживчих бізнес-процесів, що допомагає персоналу з роботи з клієнтами виконувати свої функції. Забезпечення оперативного доступу до інформації у ході контакту з клієнтом в процесі продажу та обслуговування. Охоплює маркетинг, продажі і сервіс
2. Співробітницький — програма взаємодії зі споживачами без участі персоналу з роботи з клієнтами. Забезпечує безпосередню участь клієнта в діяльності фірми і можливість впливати на процеси розробки продукту, його виробництво, сервісне обслуговування
3. Аналітичний — аналіз інформації про споживачів із різноманітними цілями. Спільний аналіз даних, що характеризують діяльність клієнта і фірми. Отримання нових знань, висновків, рекомендацій і т. д. Використовує складні математичні моделі для пошуку статистичних закономірностей і вибору найефективнішої стратегії маркетингу, продажів, обслуговування клієнтів[1]

					ІА351.070БАК.001 ПЗ	Аркуш
						10
Зм	Арк.	№ документа	Підпис	Дата		

### Принципи CRM-систем:

1. наявність єдиного сховища інформації, звідки в будь-який момент доступні усі відомості про усі випадки взаємодії з клієнтом;
2. синхронізація управління множинними каналами взаємодії;
3. постійний аналіз зібраної інформації про клієнтів та прийняття відповідних організаційних рішень — наприклад, «сортування» клієнтів на основі їхньої значимості для компанії.[1]

### Можливості

1. Швидкий доступ до актуальної інформації про клієнтів;
2. Оперативність обслуговування клієнтів та проведення операцій;
3. Формалізація схем взаємодії з клієнтами, автоматизація документообігу;
4. Швидке отримання всіх необхідних звітних даних та аналітичної інформації;
5. Зниження операційних витрат менеджерів;
6. Контроль роботи менеджерів;
7. Узгоджена взаємодія між співробітниками і підрозділами.
8. Управління бізнес-процесами — дозволяє автоматизувати послідовні операції, які виконуються співробітниками організації;
9. Управління контактами, історія взаємодії з клієнтами — це єдина база даних всіх контрагентів компанії (клієнтів, постачальників, конкурентів) з внесеною раніше докладною інформацією про них, про їх співробітників тощо. Система дозволяє здійснювати швидкий пошук важливої інформації про контрагентів, отримувати всю історію зустрічей, переговорів, листування, угод та інше. Це дуже зручний інструмент для швидкої і якісної роботи з величезними масивами інформації про клієнтів. Система автоматично нагадує про необхідність зробити дзвінок, про заплановані зустрічі та інші заходи;

					<i>IA351.070BAK.001 ПЗ</i>	Аркуш
						11
Зм	Арк.	№ документа	Підпис	Дата		

- 10.Планування та управління продажами — CRM дозволяє складати плани за різними показниками (дохід з продажу по менеджерам, відділам, продуктам). По історії проектів можна відбудувати воронку продажів, що дозволяє визначати проблемні зони в циклах продажів. Планування і контроль виконання плану по факту. Є можливість ведення різних прайс-листів (оптових, дрібнооптових, роздрібних), враховувати акційні пропозиції, знижки від обсягу покупки. Вся робота з клієнтом відбувається в одній системі: планування заходів, здійснення угод, підготовка і виписка необхідних звітних документів;
- 11.Планування та управління закупівлями і доставками — в системі менеджери завжди можуть бачити наявність і кількість товарів на складі. Відповідальні співробітники можуть стежити виконанням плану закупівель;
- 12.Управління маркетингом — електронна розсилка, пряма розсилка, sms розсилання. Система дозволяє управляти маркетинговими заходами і визначати їхню результативність. Можливість сегментації наявних в базі клієнтів (діючих і потенційних) за певними параметрами для проведення маркетингових заходів;
- 13.Автоматизація документообігу — в систему можна ввести шаблони будь-яких документів, які використовуються в організації, при цьому зникає необхідність ручного складання нового документа при виникненні події. Швидке автоматичне заповнення шаблонів договорів, які зберігаються в системі. Автоматичне виставлення рахунків і контроль оплати по них через сумісність з Клієнт-банком;
- 14.Можливість роботи через мережу;
- 15.Імпорт контрагентів з інших баз;

### 1.1.1 Класифікація CRM-систем

За даними статистики відомо, що близько 65% підприємців інтегрують CRM в підприємство протягом перших років після відкриття компанії. Однак, підприємства в різних галузях та різного масштабу мають різні потреби та вимоги до CRM, яку вони використовують. Наприклад, менеджерам крупного банку важливо отримувати аналітичні звіти по тисячам операцій клієнтів, а невеликому підприємству потрібно просто зафіксувати клієнтів до спеціалістів, наповнити їх візитами та фіксацією оплати. Тому існують різні типи CRM-систем, які відрізняються рівнем обробки інформації і завданнями, які вони вирішують.[2]

### 1.1.2 Операційні CRM

Більшість існуючих рішень відноситься до цього типу. Такі системи спрощують взаємодію з клієнтами, систематизують дані про заявки і угодах, самі виставляють рахунки, нагадують передзвонити клієнтові і можуть самі відправити йому sms-повідомлення, записують телефонні дзвінки і т.д.

Головне завдання операційних CRM-систем - підвищити лояльність клієнта в процесі безпосереднього контакту з ним.

Всі операційні CRM вміють:

- реєструвати вхідний трафік (дзвінки, листи, заявки на сайт);
- зберігати в базі дані про клієнтів, заявках, угодах, завдання і т.
- автоматизувати документообіг всередині компанії;
- фіксувати просування угод по воронці продажів;
- нагадувати про заплановані дзвінки, листах, зустрічах;
- ставити завдання і контролювати роботу співробітників.

Прикладами операційних CRM-систем можуть служити рішення 1С: CRM і БІТ: CRM 8. Однак в «чистому» вигляді операційні CRM-

					<i>IA351.070BAK.001 ПЗ</i>	Аркуш
						13
Зм	Арк.	№ документа	Підпис	Дата		

рішення сьогодні зустрічаються нечасто. Все більше розробників додають в свої програми аналітичні і статистичні функції.

### 1.1.3 Аналітичні CRM

На відміну від операційних систем, аналітичні не тільки фіксують історію взаємодії з клієнтом, але і допомагають простежити закономірності в продажах: клієнти з яких джерел купують найчастіше, на якому етапі зривається більшість угод, як розподілені клієнти по воронці продажів - і всі ці дані оновлюються в онлайн-режимі, в розрізі кожного параметра.

Мета аналітичних CRM - це аналіз накопиченої інформації про клієнтів і продажах для вибудовування більш ефективної стратегії.

Аналітичні CRM вміють:

- сегментувати клієнтську базу;
- визначати цінність клієнта;
- аналізувати їх рентабельність;
- моніторити поведінку клієнтів на кожному етапі операції;
- показувати розподіл угод по воронці продажів;
- аналізувати динаміку продажів;
- аналізувати ефективність маркетингових інструментів;
- прогнозувати обсяг продажів.

Прикладами аналітичних CRM можуть служити продукти від компанії SAS, SAP BusinessObjects і Marketing Analytic.

### 1.1.3 Колабораційні CRM

Такі CRM налагоджують комунікації з клієнтами для збору зворотного зв'язку. Інформація, отримана з їх допомогою, допомагає скорегувати асортимент товарів, цінову політику, а також процес обслуговування покупців. Наприклад, співробітники call-центру

					<i><b>IA351.070БАК.001 ПЗ</b></i>	Аркуш
						14
Зм	Арк.	№ документа	Підпис	Дата		

автосалону обдзвонюють клієнтів, задаючи питання про якість сервісу і фіксуючи відповіді. За результатами опитування закуповуються відсутні комплектуючі та додаються нові сервісні послуги.

Як таких готових CRM-систем цього типу немає: розробляються або індивідуальні рішення, або використовуються існуючі канали зв'язку, і інформація з них фіксується в основній CRM-програми (інтернет-форуми, соцмережі, телефонія, e-mail-листування).

#### 1.1.4 Комбіновані CRM-системи

В останні роки на передній план виходять програми, що поєднують в собі елементи різних типів CRM. В основному це операційні CRM з набором аналітичних функцій (звіти з продажу, клієнтам, ефективності менеджерів і т.д.), а також можливістю спілкування з клієнтами (завдяки інтеграції з сайтом, месенджерами, соцмережами і т.д.).

Саме комбіновані системи сьогодні знаходяться в ТОПі популярності малого і середнього бізнесу України та світу. До них можна віднести «Мегаплан», AmoCRM, Бітрікс24, а також інші програми: SalesapCRM, FreshOffice, RetailCRM і т.д.

### 1.2 Salesforce CRM

Salesforce Sales Cloud є популярним CRM-продуктом, який включає окремі рішення для хмари спільноти, сервісу, маркетингу та аналітики. Система допомагає відстежувати всі взаємодії з клієнтами та інформацію в одному місці, а також стежити за потенційними клієнтами.

Переваги.

1. Швидкий старт. Вам не потрібно встановлювати програмне забезпечення. Також немає апаратних вимог. Salesforce надає клієнтам унікальну можливість запустити свій бізнес всього за пару

					<b>IA351.070БАК.001 ПЗ</b>	Аркуш
						15
Зм	Арк.	№ документа	Підпис	Дата		

- днів. Розміщення в хмарі забезпечує доступ до даних у реальному часі та підвищує ефективність бізнесу, мінімізуючи час відповіді [6].
2. Автоматизація бізнесу. Salesforce CRM дозволяє зробити бізнес більш ефективним. Менеджери з продажу можуть уважно стежити за клієнтами та угодами, детально аналізувати усі данні та коригувати стратегію продажів відповідно до отриманої інформації. Також, система дає багато можливостей для автоматизації процесів – те, що раніше робили робітники компанії, система зробить автоматично, потрібно тільки налаштувати її.
  3. Гнучкість. Дизайн CRM залежить від конкретних потреб бізнесу і може бути скоригований для кожного конкретного бізнес-кейсу. Вона може бути легко налаштована будь-яким адміністратором, це не потребує програмістських навичок. Безумовно, впровадження такої інтелектуальної системи полегшує пошук нових потенційних клієнтів і прискорює автоматизацію внутрішніх процесів продажу. Саме через гнучкість цієї системи, та можливість розробки додатків до неї, ця CRM була вибрана для інтеграції системи аналізу та обробки даних.
  4. Можливість розробки персональних додатків. CRM дає можливість створити будь-який додаток за допомогою її власного фреймворку. Таким чином, навіть найвибагливіші клієнти будуть задоволені, коли спеціалісти розроблять додатки найбільш актуальні та потрібні для покращення функціонування підприємства.
  5. Мобільні додатки. Постачальник пропонує також мобільні програми для пристроїв iOS і Android із спеціальними звітами та даними в реальному часі з інформаційних панелей.
  6. Безкоштовні додатки від інших користувачів. Також, клієнти CRM мають можливість розширити функціональність, додавши готові додаткові інструменти з AppExchange (у тому числі безкоштовні).

					<i>IA351.070БАК.001 ПЗ</i>	Аркуш
						16
Зм	Арк.	№ документа	Підпис	Дата		



7. Можливість спробувати функціонал системи безкоштовно. Компанія пропонує великий безкоштовний план усім потенційним клієнтам, який дозволяє їм спочатку спробувати всі ключові функції.

#### Недоліки.

1. Ціна. Для малого бізнесу ця CRM є занадто дорогою. Ціна за одного користувача коливається від 5\$ до 250\$ за місяць (дивлячись на вибрані для використання внутрішні продукти, об'єм даних та інші параметри).
2. Складність. Через величезну кількість функцій та можливостей, які пропонує ця система, звичайні користувачі потребують тренувань та певний час, щоб звикнути до інтерфейсу. Але це не унікальна проблема Salesforce, зазвичай кожна компанія-користувач CRM має проводити спеціальні тренінги для співробітників для здобуття ними необхідних навичок користування CRM. Та Salesforce дбає про своїх користувачів, тож було створено спеціальний портал для вивчення платформи та здобуття навичок користування її функціями, яке проходить легко та в форматі гри. Портал може бути використано як звичайними користувачами, так і адміністраторами, що налаштовують CRM в залежності від потреб підприємства, та навіть програмістам, що створюють додатки для конкретних потреб компанії-користувача CRM [5].

					ІА351.070БАК.001 ПЗ	Аркуш
						17
Зм	Арк.	№ документа	Підпис	Дата		

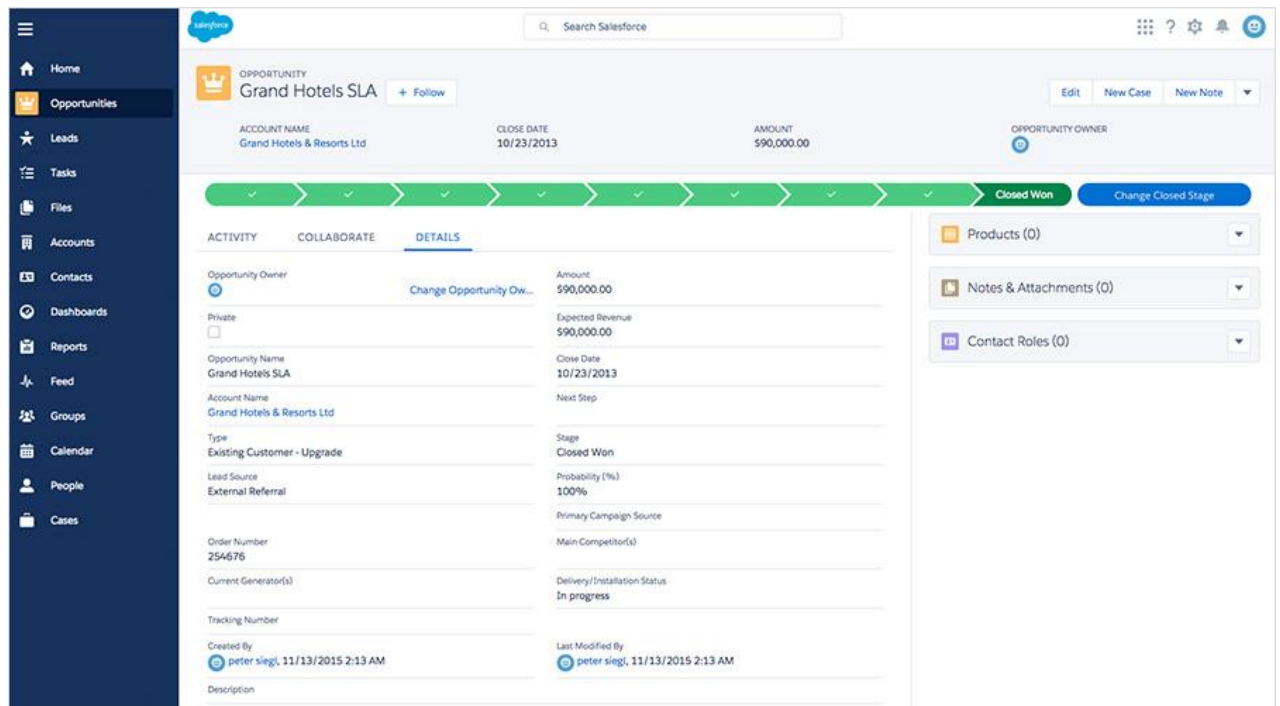


Рисунок 2.3.1 – Приклад інтерфейсу Salesforce CRM

### 1.3 HubSpot CRM

Hubspot CRM - це просте і безкоштовне рішення, яке дозволяє командам продажів використовувати існуючий робочий процес без значних змін, щоб швидко розпочати роботу.[28]

Переваги.

1. Програма доступна безкоштовно, тому немає ризику спробувати її, щоб побачити, як вона працює. Це одне з найкращих безкоштовних рішень CRM. Згідно з HubSpot, CRM буде вільно використовувати назавжди - навіть для нових клієнтів HubSpot. Він не тільки безкоштовний, але включає в себе необмежену кількість користувачів - це означає, що компанії-користувачі можуть додати усю команду - маркетинг, продаж, підтримка, ін. Однак, безкоштовна версія все ж має ліміти, які можуть суттєво заважати великому бізнесу, тож вони матимуть придбати платну версію системи щоб задовільнити їх потреби.

Зм	Арк.	№ документа	Підпис	Дата

IA351.070БАК.001 ПЗ

Аркуш

18

2. Кращі ідеї. Система HubSpot CRM автоматично відстежує взаємодію з клієнтами, надаючи цінну інформацію про поведінку клієнтів і дає знати, коли їм слід приділити більше уваги.
3. Використовує ту саму базу даних, що й маркетингова платформа (не потрібна інтеграція). HubSpot CRM зразу пов'язаний з маркетинговими інструментами HubSpot, що дозволяє плавно переходити від маркетингу до продажів у налаштованій послідовності і значно полегшити передачу даних.
4. Можна використовувати з іншими CRM. CRM HubSpot не впливає на те, що організація здійснює в інших платформах CRM, таких як SFDC або Microsoft Dynamics. Їх можна використовувати одночасно - і обидва можуть бути інтегровані в HubSpot.

#### Недоліки.

1. Відносно новий. Оскільки HubSpot CRM все ще є відносно новим у порівнянні з іншими основними гравцями CRM, існують деякі обмеження функцій. Однак акцент HubSpot на дослідженнях і розробках у 2018 році пом'якшив багато колишніх сумнівів щодо платформи, і регулярно проводяться оновлення.
2. Простий. Хоча HubSpot CRM відмінно підходить для зростаючих підприємств, організації з великими групами продажів можуть виявити, що HubSpot CRM поки що недостатньо розвинений для їх потреб.
3. Відсутність гнучкості. У порівнянні з іншими CRM, платформа HubSpot є відносно негнучкою. Хоча, звичайно, є особливості, які можна налаштувати навколо процесів і потреб команди, було виявлено, що організації з дуже складними процесами продажу не можуть знайти гнучкість, яку вони потребують у HubSpot CRM.

					<i>IA351.070БАК.001 ПЗ</i>	Аркуш
						19
Зм	Арк.	№ документа	Підпис	Дата		

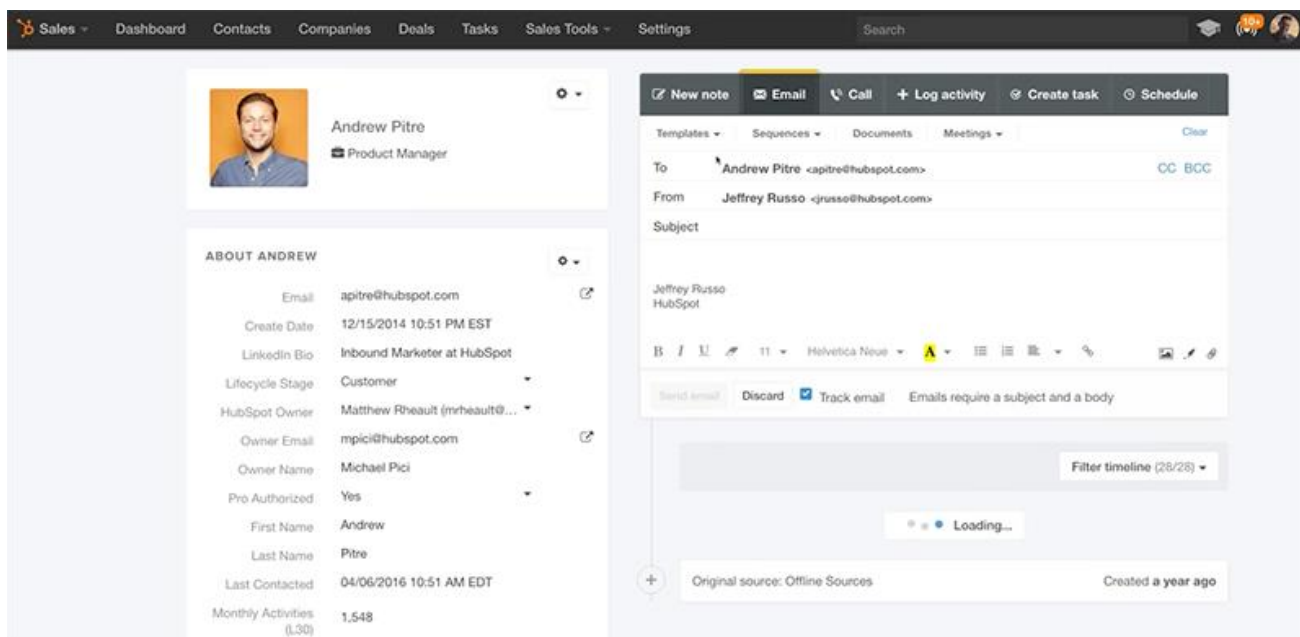


Рисунок 2.3.2 – Приклад інтерфейсу HubSpot CRM

## 1.4 Microsoft Dynamics CRM

Microsoft Dynamics CRM розроблений для того, щоб допомогти компаніям ринку більш розумно, і продавати продуктивно і ефективно. Система пропонує корисні функції, включаючи бізнес-аналітику, соціальні ідеї та продуктивність, використовуючи програми One Microsoft. Клієнти можуть вибрати розгортання рішення на локальному ПК або мережі, в хмарі або використовувати гібридну комбінацію.

До переваг Microsoft Dynamics CRM (рис. 2.3.3) належать: інтегровані засоби автоматизації та маркетингу, численні інформаційні панелі, тісна інтеграція з додатками Microsoft, автономний доступ та безшовна інтеграція з системами та програмами третіх сторін.

Переваги.

1. Налаштування (робочі процеси, перегляди тощо) легко створювати та управляти.

Зм	Арк.	№ документа	Підпис	Дата

IA351.070БАК.001 ПЗ

Аркуш

20

2. Гнучкість (онлайн або на місці), простота використання, відкрита платформа для розширених налаштувань і конфігурацій, зниження вартості ліцензування та хороша підтримка від Microsoft.
3. Інтеграція з Outlook є зручною для більшості користувачів. Вона також має дуже повний набір опцій, які можна налаштувати для більшості підприємств.
4. Динаміка – багатофункціональний додаток, що дає можливість аналізувати дані з метою використання при плануванні бізнесу.

#### Недоліки.

1. Доступність є проблемою для малих компаній.
2. Як і будь-який продукт корпорації Майкрософт, помилки можуть бути несподіваними та важкими для виправлення.
3. Досить важкий процес впровадження.
4. Ціни значно збільшуються, оскільки компанії зазвичай додають більше користувачів або потребує більше можливостей, тож клієнти повинні мати на увазі те, що їм потрібно з самого початку і заздалегідь знати або мати уявлення про те, як використовувати його, щоб витрати не були неочікуваними, та прибуток від використання CRM перевищував витрати на неї.[29]

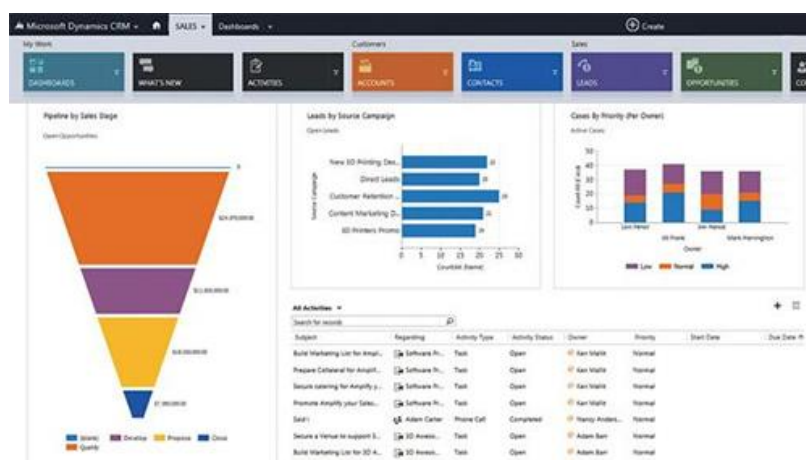


Рисунок 2.3.3 – Приклад інтерфейсу Microsoft Dynamics CRM

## 1.5 Bitrix24

Bitrix24 - це спільне програмне забезпечення, призначене для автоматизації процедур зв'язку та управління в малих і великих підприємствах або навіть стартапах. Цей інструмент пропонує уніфікований і повний набір інструментів для керування контактами, повідомленнями, електронною поштою, файлами, завданнями та проектами. Ця CRM є однією з найбільш популярних в Україні.[30]

### Переваги.

1. Великий спектр можливостей. Система істотно спрощує планування завдань і ведення клієнтської бази, дає можливість забезпечити інтеграцію з інтернет-магазином і автоматизувати бізнес-процес;
2. Вбудована смарт-телефонія.
3. Можливість інтеграції з 1С. Це дуже важливий параметр для великої кількості компаній в Україні. Правильна настройка Бітрікс24 допомагає вирішувати масу;
4. Це хмарне програмне забезпечення.
5. Простота. Працювати з системою легко, а процес навчання рядового співробітника займає мало часу, а також є можливість швидкого пошуку потрібної інформації.
6. Підтримує веб-і мобільні програми, які підходять для пристроїв Android, iPhone або iPad. Але деякі можливості доступні тільки для користувачів платної версії CRM.
7. Має безкоштовну версію, але вона обмежена лише 12 користувачами, тож використовується потенційними клієнтами здебільшого для ознайомлення з функціями та можливостями CRM перед придбанням повної версії (є декілька планів в залежності від кількості користувачів) [30]

					ІА351.070БАК.001 ПЗ	Аркуш
						22
Зм	Арк.	№ документа	Підпис	Дата		

Недоліки.

1. Необхідність роботи в рамках встановленого розробниками функціоналу. По частині налаштувань система замкнута, так що розширити поля даних клієнтів або провести подібні роботи не так просто;
2. Необхідність звикання до інтерфейсу. Багато користувачів відзначають, що далеко не відразу вдається зорієнтуватися у всьому багатстві можливостей. У Бітрікс24 досить складна візуалізація і не завжди буває просто відшукати потрібну задачу серед завершених.

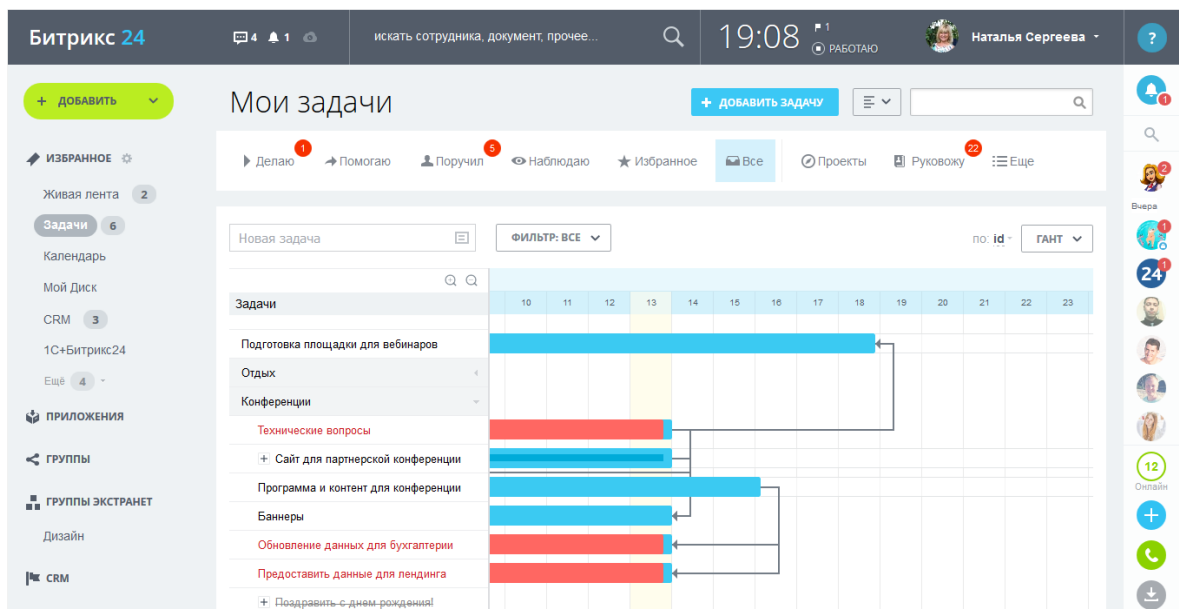


Рисунок 2.3.4 – Приклад інтерфейсу Бітрікс24 CRM

## 1.6 Висновки

Було уважно проаналізовано можливості, переваги та недоліки зазначених вище CRM, а також ряду інших, і після ретельних тестів зроблено висновок, що Salesforce CRM є найбільш зручною для реалізації додатку для аналізу та обробки даних з усіх популярних рішень, ця система дає багато можливостей для розробки персональних додатків, фреймворк надає широкий вибір інструментів для

програмування та інтеграції з сторонніми системами, тож існує можливість створення системи аналізу та обробки даних на основі цієї CRM системи, що є вирішальним фактором при виборі CRM.

Salesforce є однією з найбільш популярних CRM систем, та продовжує набирати популярність. Причинами на це є її надзвичайна гнучкість, можливості налаштування її під будь-яке підприємство, великий вибір функцій для використання та регулярні оновлення тричі на рік. Незаперечною перевагою SalesForce CRM є те, що це хмарний сервіс, тож не потребує встановлення нового програмного забезпечення, оновлення комп'ютерів та купівлі серверів для зберігання даних – усе це відбувається в хмарі, а користувачу потрібен тільки доступ в інтернет та будь-який браузер на вибір.

Недоліком Salesforce CRM можна назвати хіба що її ціну, вона є відносно дорогою, не усі підприємці в Україні можуть дозволити собі використання цієї CRM, але це не заважає їй користуватися популярністю в Європі та Америці.

Salesforce надає можливість користуватися безкоштовною версією CRM, хоча вона і має деякі обмеження щодо кількості користувачів та об'єму даних, що можуть зберігатися в хмарній базі даних, та ці обмеження не є критичними для розробки та демонстрації роботи додатку.

					ІА351.070БАК.001 ПЗ	Аркуш
						24
Зм	Арк.	№ документа	Підпис	Дата		



## 2. ДОСЛІДЖЕННЯ ПРОБЛЕМИ ПРОГНОЗУВАННЯ

### 2.1 Загальні відомості

Прогноз в бізнесі важливий через можливе використання його для ефекту стабілізації. Розумні прогнози спонукають людей діяти більш раціонально і попереджають їх «сверхреакцію» в бік песимізму або оптимізму. Хороший прогноз забезпечує фірмі прийняття раціональних рішень щодо вироблених фірмою товарів або послуг. Відсутність прогнозу змушує керівництво фірми робити зайві запобіжні заходи.[3]

Методи прогнозу зазвичай вимагають великих витрат часу і грошей. Однак підприємець потребує методи, які не вимагають складних умовиводів в повсякденній роботі і можуть бути представлені у вигляді програм. Необхідно знайти методи прогнозування без детального індивідуального аналізу. До того ж бажано, щоб знання ситуації на ринку, якими володіють люди, які постійно працюють на ньому, були використані в подібних моделях.

Оскільки прогнозування є важкою проблемою, то очевидно, що фірма повинна мати кілька серій прогнозів, відмінних від простого описового прогнозу. Це допоможе приймати більш рішучі дії, результатом яких є зростання прибутку, підвищення ефективності роботи організації і зростання її престижу.[3]

Вихідні дані для складання прогнозу з використанням часових рядів зазвичай представляють собою результати вибірових спостережень змінних - або інтенсивності (наприклад, попит на продукцію), або стану (наприклад, ціна). Рішення, які повинні прийматися в даний момент, визначаються в подальшому після деякого проміжку часу, величина якого може бути прогнозованою.

Тимчасові ряди являють собою впорядковані в часі дані. Відповідно до цього ми будемо надалі позначати період часу через  $t$ , а відповідне йому значення даних через  $y(t)$ . Відзначимо, що членами

					ІА351.070БАК.001 ПЗ	Аркуш
						25
Зм	Арк.	№ документа	Підпис	Дата		

часового ряду є або суми, або числова інформація, отримана в певний момент часу. Наприклад, сума тижневих продажів в магазині, що отримується в кінці кожного тижня протягом року, утворює тимчасовий ряд.[4]

Тренд означає загальний напрям і динаміку тимчасового ряду. У цьому визначенні наголос робиться на понятті «загальний напрямок», оскільки основну тенденцію необхідно відокремити від короткострокових коливань, що представляють собою циклічні і сезонні коливання. [4] Приклади циклічних коливань: ціни на промислову сировину, курси акцій, обсяги продажів в оптовій і роздрібній торгівлі та ін. Сезонні коливання зустрічаються в тимчасових рядах, що описують продаж, виробництво, зайнятість і ін. Важливу роль в сезонних коливаннях відіграють погодні умови, мода, стиль і т. д. Особливо відзначимо, що нерегулярні або випадкові коливання часових рядів не підкоряються ніякої закономірності і не існує теорії, здатної передбачити їхню поведінку.

З точки зору вироблення правильного рішення керівництвом фірми, включення періодичних (циклічних і сезонних) коливань в загальну модель може підвищити ефективність прогнозу і дозволить передбачити очікувані високі і низькі значення прогнозованих змінних. При цьому потрібно мати на увазі, що «ділові» або економічні цикли не можна відтворити з точністю, що дозволяє на практиці робити висновки про майбутні злети та спади, виходячи з аналізу минулого.[4]

Основна мета економічного аналізу - отримання найбільшого числа ключових параметрів, що дають об'єктивну картину фінансового стану підприємства, його прибутків і збитків, змін в структурі активів і пасивів. Економічний аналіз дозволяє виявити найбільш раціональні напрямки розподілу матеріальних, трудових і фінансових ресурсів.[4]

					ІА351.070БАК.001 ПЗ	Аркуш
						26
Зм	Арк.	№ документа	Підпис	Дата		

Можна виділити наступні основні принципи аналізу і обробки даних:

1. Науковість - базується на положеннях динамічної теорії пізнання, враховувати вимоги економічних законів, використовувати досягнення науково-технічного прогресу, а також методи економічних досліджень. Принцип науковості реалізується шляхом вдосконалення аналізу господарської діяльності, застосування методик і ЕОМ.[3]
2. Об'єктивність, конкретність і точність - передбачає дослідження реальних економічних явищ і процесів і їх причинно-наслідковий зв'язок. Він повинен базуватися на достовірній, перевірній інформації, а вигода його повинна обґрунтовуватися точними аналітичними розрахунками. З цієї вимоги випливає необхідність постійного вдосконалення організації обліку, внутрішнього і зовнішнього аудиту, а так само методики аналізу з метою підвищення його точності і достовірності розрахунку.
3. Системність і комплексність - кожен досліджуваний об'єкт розглядається як складна динамічна система, що складається з ряду елементів, певним чином пов'язаних між собою. Так само вивчення кожного об'єкта повинно здійснюватися з урахуванням всіх внутрішніх і зовнішніх зв'язків, взаємозалежності його окремих елементів, певним чином пов'язаних між собою. Вивчення кожного об'єкта має здійснюватися з урахуванням всіх внутрішніх і зовнішніх зв'язків, взаємозалежності і взаємозалікові його окремих елементів. Комплексність і дослідження вимагають охоплення всіх ланок і всіх сторін діяльності підприємств.
4. Оперативність і своєчасність - передбачає вміння швидко і чітко проводити аналіз, приймати управлінські рішення і проводити їх в життя. Оперативність аналізу полягає в своєчасному виявленні та

					ІА351.070БАК.001 ПЗ	Аркуш
						27
Зм	Арк.	№ документа	Підпис	Дата		

перерозподілі причин відхилення від плану, як за кількісними, так і за якісними показниками, пошук шляхів усунення негативно-діючих факторів і закріплення посилення позитивних факторів. Все це робить можливим поліпшення роботи підприємств.

5. Дієвість - активний вплив на хід процесу виробництва і його результатів.
6. Плановість і систематичність - аналіз проводиться за планом і періодично. Цей принцип дозволяє планувати роботу.
7. Демократичність - припускає участь всіх в проведенні аналізу та передбачає доступність інформації всім. Хто приймає рішення.
8. Ефективність - витрати на його проведення повинні давати багаторазовий ефект.

Основними функціями фінансового аналізу є:

- об'єктивна оцінка фінансового стану, фінансових результатів, ефективності та ділової активності аналізованої компанії;
- виявлення факторів і причин досягнутого стану та отриманих результатів;
- Підготовка та обґрунтування прийнятих управлінських рішень в галузі фінансів;
- виявлення і мобілізація резервів поліпшення фінансового стану і фінансових результатів, підвищення ефективності всієї господарської діяльності.

## 2.2 Причини виникнення проблеми прогнозування

Найважливішим резервом підвищення ефективності виробництва є підвищення якості прийнятих управлінських рішень. Прийняття рішень - складова частина будь-якої управлінської функції. Необхідність прийняття рішення пронизує все, що робить керівник, формуючи цілі і домагаючись їх досягнення. Удосконалення процесу прийняття обґрунтованих об'єктивних рішень в ситуаціях виняткової складності досягається шляхом використання наукового підходу до даного процесу, моделей і кількісних методів прийняття рішень.[8]

Відмінними рисами сучасного етапу практики управління є зміна і ускладнення проблем, які необхідно вирішувати керівникам різних рівнів управління; високі вимоги до якості прийнятих управлінських рішень при непостійності впливу зовнішніх факторів; необхідність миттєвого реагування на різні дестабілізуючі чинники функціонування підприємства. Ці обставини вимагають розробки нового способу управління - попереджувального в поєднанні з оперативним управлінням. В його основі лежить прогнозний блок, який виділяється в окрему функцію управління, що передує функції планування.

У сучасних умовах ринкової економіки одним з успішних методів дослідження в менеджменті є метод прогнозування. Більшість припущень, з яких виходить керівник, стосуються майбутнього, над яким керівник майже не має ніякого контролю. Однак такого роду припущення необхідні для багатьох операцій планування. Чим краще керівник зможе передбачати зовнішні і внутрішні умови щодо майбутнього, тим вище шанси на складання здійснених планів. [8]

У класичному менеджменті вважається, що прогнозування - це метод, в якому використовуються як досвід, накопичений в минулому, так і поточні припущення щодо майбутнього з метою його визначення. В результаті отримують картину майбутнього, яку можна

використовувати як основу при плануванні. Саме тому даний метод є актуальним.[12]

Для прогнозування в практичній діяльності застосовуються різні кількісні та якісні методи.

Кількісні методи (прийоми) базуються на інформації, яку можна отримати, знаючи тенденції зміни параметрів або маючи статистично достовірні залежності, що характеризують виробничу діяльність об'єкта управління. Приклади цих методів - аналіз часових рядів, каузальне (причинно-наслідкове) моделювання.

В основу якісних методів покладено експертні оцінки фахівців в області прийнятих рішень, наприклад методи експертних оцінок, думка журі (усереднення думок експертів в релевантних сферах), моделі очікування потреби.

Прогнозування сьогодні – спеціалізована галузь з підрозділами. Існують організації, які займаються тільки прогнозуванням в конкретних сферах діяльності. Прикладом може бути інститут Геллапа, який спеціалізується на зборі та аналізі інформації, що дозволяє прогнозувати переваги і результати різних політичних і соціальних процесів. Багато фірм і відділення великих підприємств проводять аналіз ринку, прагнучи спрогнозувати ставлення споживачів до запланованих нових видів продукції.[11]

### 2.3 Сучасні тенденції

З кожним роком все більше компаній приходять до висновку, що правильне використання аналітичних систем та систем обробки великих обсягів даних забезпечує істотну конкурентну перевагу на ринку. Така ситуація призводить до активного розвитку аналітичних відділень в компаніях та істотної інвестиції в інфраструктуру обробки даних.

					<b>IA351.070БАК.001 ПЗ</b>	Аркуш
						30
Зм	Арк.	№ документа	Підпис	Дата		

Відповідно до різних досліджень, щорічний приріст затрат на розвиток системного аналізу та обробки даних становить 20% - 30%. Прогнозований сумарний об'єм затрат на обробку даних та аналітичні сервіси в 2019 році склав ~ 3В \$.

Великий попит на аналітичні системи і істотні інвестиції є сильним драйвером для швидкого розвитку індустрії великих даних і поява великої кількості нових продуктів і рішень.

Відповідно до дослідженням великих консалтингових компаній, таких як Deloitte і Forrester, в 2019 році можна буде вивести наступні основні напрямки в області аналізу та обробки даних.

1. Автоматизація і коммодування процесів обробки даних
2. Розвиток аналітичних систем реального часу
3. Поява нових інструментів для аналізу даних датчиків і різних пристроїв
4. Розвиток когнітивних систем і систем автоматичного прийняття рішення
5. Зміни в законодавстві, пов'язані з зберіганням і обробкою даних

Раніше платформи бізнес-аналізу і аналітики були більш відомі відділами ІТ, тобто, більш складними в налаштуванні і менш гнучкими для кінцевого користувача, проте за останній час стався зсув в бік самодостатніх, самостійних сервісів аналізу. Більшість з того, що продається - сучасні платформи аналізу, де в основі стоїть користувач, що є новою ринковою перспективою, змінюючи позиції виробників на ринку.

В останні кілька років поступово відбувається зсув балансу від потужних засобів бізнес аналітики, в значній мірі підтримувані ІТ персоналом і забезпечують старі вимоги бізнес аналітики, до потреби в гнучкості бізнес аналізу та можливості автономної роботи користувача.

Еволюція і різноманіття автономних засобів підготовки даних, доступні можливості виявлення даних, змістили фокус покупців до простоти використання інструментів, які підтримують повний набір можливостей для процесу аналізу даних і не вимагають залучення ІТ до заздалегідь визначеним моделям даних як попередня умова для аналізу.

### 2.3.1 Автоматизація і коммодування процесів обробки даних

Більше витрат на технічне обслуговування системного зберігання і обробки даних вимусять компанії шукати більш дешеві рішення. Одним з таких рішень є відмова від власних дата-центрів і перехід на хмарні технології зберігання та аналізу даних як Amazon WS, Microsoft Azure.

Гостра проблема недостатньої кількості фахівців в області аналізу даних є хорошим стимулом для розвитку автоматизованих інструментів, які не потребують глибоких знань в області технологій і орієнтованих на рішення бізнес-задач. В якості прикладів таких інструментів можна привести Tableau, IBM Watson і Azure Machine Learning.

Окремо потрібно зазначити розвиток системної автоматичної передобробки сигнальних і неструктурованих даних подібних OpenRefine і DataWrangler.

### 2.3.2 Розвиток аналітичних систем реального часу

Традиційні аналітичні рішення на базі реляційних баз даних і технологій OLAP вимагають порівняно великої кількості часу для обробки запитів і формування звітів.

Компанія для прийняття ефективних рішень необхідно аналізувати дані в реальному часі. Це призвело до розвитку технологій потокової обробки даних і швидкого доступу до даних, що зберігаються в оперативній пам'яті. Одними з перших продуктів, створених для вирішення цих задач, є Apache Storm і Cloudera Impala.

					<i>IA351.070БАК.001 ПЗ</i>	Аркуш
						32
Зм	Арк.	№ документа	Підпис	Дата		



### 2.3.3 Поява нових інструментів для аналізу даних датчиків і різних пристроїв

Мобільна електроніка та інтернет речей продовжать активно розвиватися в 2019 році. Подібні гаджети вже знаходять широке застосування в сфері розваг і моніторингу здоров'я. Аналізуючи інформацію з цих пристроїв можна підвищити ефективність їх використання.

Широке поширення гаджетів призводить до появи нових завдань пов'язаних з аналізом сигналів різного роду датчиків і сенсорів. На відміну від транзакційних даних, дані сенсорів можуть бути зашумлені і схильні до спотворень, що в свою чергу вимагає особливого підходу до обробки подібних сигналів. Для завдань пов'язаних зі збором і аналізом інформації з датчиків будуть створені спеціальні інструменти.

Також будуть активно розвиватися стандарти щодо інтеграції датчиків і аналітичних систем, що дозволить істотно прискорити розвиток інтернету речей в цілому.

Окремо варто відзначити, що дані переносної електроніки дозволять збирати більше інформації про поведінку та вподобання користувачів, що швидше за все призведе до появи нових гравців на ринку даних.

### 2.3.4 Розвиток когнітивних систем і систем автоматичного прийняття рішення

Експерти прогнозують активний розвиток технологій в області взаємодії людини і штучного інтелекту. Зокрема, розвиток систем автоматичного прийняття рішень в бізнес процесах, систем аналізу текстової інформації, голосу, зображень та відео. Серед уже існуючих продуктів можна відзначити такі як Google Now і Word Lense.

					<i>IA351.070БАК.001 ПЗ</i>	Аркуш
						33
Зм	Арк.	№ документа	Підпис	Дата		

### 2.3.5 Зміни в законодавстві, пов'язані з зберіганням і обробкою даних

Вже сьогодні ми щодня генеруємо так багато даних, що по ним можна досить точно скласти портрет кожної людини, виявити його переваги і дізнатися особисту інформацію. З розвитком інтернету речей і появою нових гаджетів обсяг даних буде тільки збільшуватися, а разом з цим буде збільшуватися частка особистої інформації, що зберігається в дата-центрах. Наприклад, фітнес-браслети, що стали дуже популярними в останні роки, генерують і зберігають в хмарних базах даних надзвичайно великі обсяги особистої інформації користувачів (навіть такої чутливої інформації як місцезнаходження та стан здоров'я в реальному часі).

Такий розвиток ситуації призведе до підвищення соціального тиску на сервіси і компанії, що мають безпосереднє відношення до обробки подібного роду даних, що призведе до втручання регулятора і появи нових законів обмежують використання особистої інформації і визначають вимоги до безпеки зберігання даних.

Ймовірно, нові вимоги змусять переглянути архітектурні рішення компаній і приведуть до появи гібридних систем при яких частина даних замість хмари буде зберігатися в захищених локальних дата-центрах.

## 2.4 Висновки

Незважаючи на великі досягнення за попередні періоди, 2019 обіцяє бути дуже цікавим з точки зору розвитку аналітичних систем і появи нових можливостей для революційних рішень в області обробки та аналізу даних.

					<i><b>IA351.070БАК.001 ПЗ</b></i>	Аркуш
						34
Зм	Арк.	№ документа	Підпис	Дата		

### 3. РОЗРОБКА І ОПИС ДОДАТКУ АНАЛІЗУ ДАНИХ

#### 3.1 Опис алгоритму програми

Діаграма структури системи, та місце розробленого додатку в неї наведено в Додатку А. Система складається з CRM, віддаленого серверу, де проходить власне аналіз даних та прогнозування, та додатку, що забезпечує спілкування CRM та віддаленого серверу. Користувач взаємодіє виключно з інтерфейсом CRM.

Алгоритм роботи CRM з додатком наведено в Додатку Б. Алгоритм починається з запита користувача о аналізі даних в CRM. Після цього CRM передає запит, дані авторизації, налаштувань та дані угод, що мають бути проаналізовані у додаток. Коли робота додатку завершена, CRM отримує результати аналізу та прогнозування, відображує їх користувачу та повідомляє його о закінчення аналізу (так як аналіз може займати деякий час, користувач не буде чекати його закінчення на сторінці та вірогідно переключиться на інші задачі, тож система повідомляє о закінчення аналізу в електронному листі)

Блок-схема алгоритму роботи додатку наведена у Додатку В. Вона складається з декількох частин, які відбуваються у декількох середовищах:

1. Початок алгоритму відбувається в середовищі CRM-системи, коли користувач бажає розпочати аналіз даних (або якщо він налаштував автоматичний початок аналізу на певний час, і цей час настав). Тоді система відправляє HTTP запит на сервер, де буде відбуватися аналіз та прогнозування.
2. Коли запит успішно прийнятий на сервері, контролер перевіряє валідність токenu авторизації, що був переданий як один з параметрів HTTP запиту. Якщо токен діючий, програма переходить до аналізу, в іншому разі вона відправляє

повідомлення назад до CRM системи, яка відображає деталі помилки користувачу і алгоритм зупиняється.

3. Далі система робить запит авторизації до Salesforce CRM з метою отримання даних та налаштувань, що там зберігаються. Лістинг програми наведено у Додатку Е.
4. Після цього система завантажує усі дані, що мають бути аналізовані з Salesforce CRM, а також персональні налаштування користувача та компанії. Лістинг програми наведено у Додатку З.
5. Застосовується алгоритм Random Forest для створення «лісу» - великої кількості дерев рішень, що створюються, навчаються та випробовуються на даних, завантажених з CRM (більш детально, даних об угодах, що вже є закритими – програні чи виграні).  
Блок-схема алгоритму аналізу даних Random Forest наведено в Додатку Г. Окремо наведено алгоритм навчання класифікатора «створення лісу» у Додатку Д. Лістинг програми наведено у Додатку Ж.
6. Створений «ліс» застосовується для прогнозування результату усіх відкритих угод (шляхом голосування усіх дерев). Лістинг програми наведено у Додатку Ж.
7. Результати прогнозування відправляються назад до Salesforce CRM для відображення користувачу. Лістинг програми наведено у Додатку З.
8. Користувач проінформований про закінчення аналізу через електронний лист.

У наступному розділі більш детально розглянуто саме алгоритм аналізу та прогнозування Random Forest.

### **3.2 Опис алгоритму аналізу даних Random Forest**

Для розробки системи аналізу та обробки даних я вибрала алгоритм Random Forest.

					<b>IA351.070БАК.001 ПЗ</b>	Аркуш
						36
Зм	Арк.	№ документа	Підпис	Дата		

Random forest (англ. Випадковий ліс) - алгоритм машинного навчання, запропонований Лео Брейманом і Адель Катлер, що полягає у використанні комітету (ансамблю) вирішальних дерев для тренування моделі, які потім використовуються для класифікації та прогнозу. Алгоритм поєднує в собі дві основні ідеї: метод беггінга Брейман, і метод випадкових підпросторів, запропонований Tin Kam Ho. Алгоритм застосовується для задач класифікації, регресії і кластеризації. Недоліком є схильність до перенавчання.

### Опис алгоритму Random Forest

Random Forest є методом машинного навчання на основі дерев приймання рішень з використанням беггінгу, на кожному кроці ітерації він розбиває множину даних на підмножини за певною ознакою. Почати опис алгоритму випадкового лісу слід з дерева прийняття рішень, як з основного структурного елементу лісу, адже саме від того, яким чином побудовано кожне дерево, серйозно залежить якість роботи і стійкість всієї фінальної композиції.[9]

### Дерево прийняття рішень

По суті, дерево рішень поділяє початкові дані на менші групи даних на основі певної ознаки цих даних (поля даних), доки у нас не буде достатньо підгруп для приймання конкретного рішення. Давайте розглянемо приклад. Нижче наведено дерево рішень, чи варто грати в теніс (рис. 4.1.1).

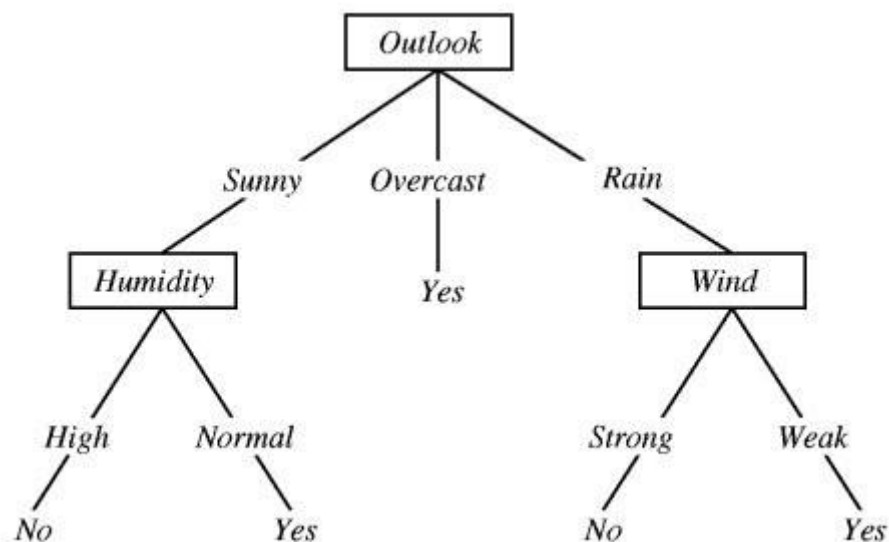


Рисунок 4.1.1 – Приклад дерева приймання рішень

У наведеному вище прикладі дерево рішень розбивається на декілька «гілок», поки ми не дійшли висновку «Так», ми повинні грати в теніс, або «Ні» ми не повинні грати в теніс. Дотримуйтесь рядків уздовж дерева, щоб визначити рішення. Наприклад, якщо прогноз похмурий, то "Так", ми повинні грати в теніс. Якщо прогноз сонячний і вологість висока, то «ні» ми не повинні грати в теніс.

У моделі дерева рішень ці розбиття вибираються відповідно до міри чистоти. Тобто, на кожному вузлі ми хочемо, щоб інформаційний приріст був максимізований.

Bootstrapping - це методика вибірки, в якій ми випадково вибираємо заміну з набору даних. При Bootstrapping ми використовуємо лише близько 2/3 даних. Приблизно 1/3 даних (дані “ out-of-bag”) не використовуються в моделі і можуть зручно використовуватися в якості тестового набору.

Беггінг, або бутстрап-агрегація, - це створення набору дерев, кожен з яких навчений на X бутстрап-наборах. Остаточне передбачене значення – середнє значення всіх X дерев рішень. Одне єдине дерево

Зм	Арк.	№ документа	Підпис	Дата

рішень має високу дисперсію (прагне до перенасичення), тому, беггінг поєднує багатьох «слабких учнів» у декількох «сильних учнів», в результаті, ми усереднюємо відхилення (рис. 4.1.2).

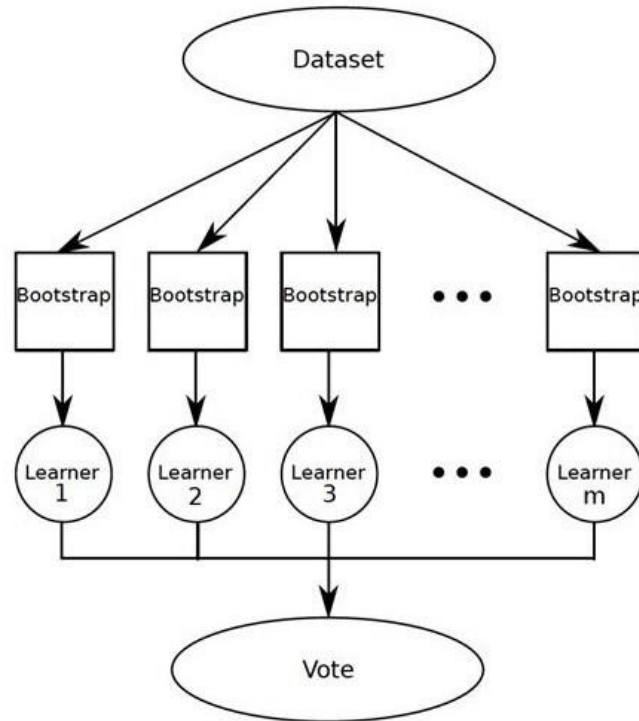


Рисунок 4.1.2 – Принцип роботи бутстрап-агрегації, або беггінгу

Random forest поліпшується в результаті беггінгу, оскільки він об'єднує розбиття даних за випадковою підмножина ознак з бутстрап-агрегацією. Це означає, що при кожному розбитті дерева модель розглядає лише невелику підмножину ознак, а не всі ознаки моделі. Тобто, з набору доступних ознак  $n$  підмножина  $m$  ознак ( $m = \sqrt{n}$ ) вибирається випадковим чином. Це важливо, щоб дисперсія була усереднена. Розглянемо, що станеться, якщо набір даних містить декілька сильних предикторів. Ці предиктори послідовно вибиратимуться на верхньому рівні дерев, тому у нас будуть дуже схожі структуровані дерева. Іншими словами, дерева будуть сильно корельовані.[10]

Таким чином, алгоритм Random Forest – це модель дерев рішень з використанням беггінгу на кожному підмножині ознак кожного розбиття.

#### Алгоритм навчання класифікатора

Нехай навчальна вибірка складається з  $N$  прикладів, розмір простору ознак дорівнює  $M$ , і заданий параметр  $m$  (в задачах класифікації зазвичай  $m \approx \sqrt{M}$ ).

Всі дерева комітету будуються незалежно один від одного за такою процедурою:

1. Генерується випадкова підвибірка з повторенням розміром  $N$  з навчальної вибірки. (Таким чином, деякі приклади потраплять в неї кілька разів, а приблизно  $N / 3$  прикладів не ввійдуть в неї взагалі).[9]
2. Будується вирішальне дерево, що класифікує приклади такої підвибірки, причому в ході створення чергового вузла дерева вибирається ознака, на основі якої проводиться розбиття, не з усіх  $M$  ознак, а лише з  $m$  випадково обраних. Вибір найкращої з цих  $m$  ознак може здійснюватися різними способами. В оригінальному коді Брейман використовується критерій Гіні, що застосовується також в алгоритмі побудови вирішальних дерев CART. У деяких реалізаціях алгоритму замість нього використовується критерій приросту інформації. [10]
3. Дерево будується до повного вичерпання підвибірки і не піддається процедурі прунінга (на відміну від вирішальних дерев, побудованих за таким алгоритмам, як CART і ID3).[9]

Класифікація об'єктів проводиться шляхом голосування: кожне дерево комітету відносить класифікуємий об'єкт до одного з класів, і перемагає клас, за який проголосувала найбільша кількість дерев.

					<i>IA351.070BAK.001 ПЗ</i>	Аркуш
						40
Зм	Арк.	№ документа	Підпис	Дата		



Оптимальне число дерев підбирається таким чином, щоб мінімізувати помилку класифікатора на тестовій вибірці. У разі відсутності помилки класифікатора, мінімізується оцінка помилки out-of-bag: частка прикладів навчальної вибірки неправильно класифікується комітетом, якщо не враховувати голоси дерев на прикладах, що входять в їх власну навчальну підвибірку.

#### Переваги

1. Висока якість одержуваних моделей, порівнянне з SVM і бустінгом, і краще, ніж у нейронних мереж.
2. Універсальність. Алгоритм є надзвичайно універсальним, незалежно від того, чи вирішується завдання регресії або класифікації, випадкові ліси є придатною моделлю. Він може працювати з двійковими функціями, категоріальними особливостями та числовими функціями. Необхідно дуже мало попередньої обробки даних, яку потрібно зробити, перед тим, як застосовувати даний алгоритм. Дані не потребують повторного масштабування або перетворення.
3. Здатність ефективно обробляти дані з великим числом ознак і класів.
4. Нечутливість до масштабування (і взагалі до будь-яких монотонних перетворень) значень ознак.
5. Однаково добре обробляються як безперервні, так і дискретні ознаки. Існують методи побудови дерев за даними з пропущеними значеннями ознак.
6. Існують методи оцінювання значущості окремих ознак в моделі.
7. Алгоритм Random Forest чудово справляється з багатовимірними даними, оскільки працює з підмножинами даних.
8. Існують методи оцінювання значущості окремих ознак в моделі.
9. Внутрішня оцінка здатності моделі до узагальнення (тест out-of-bag).
10. Здатність працювати паралельно в багато потоків. Процес обробки з цим алгоритмом може бути розділений на кілька машин для запуску.

Це призводить до швидшого часу обчислення. Посилені моделі, навпаки, є послідовними, і їх обчислення триватиме довше.

11. Швидкість. Алгоритм забезпечує відносно швидкий прогноз. Він швидше навчається, ніж дерева рішень, оскільки працює тільки на підмножині ознак цієї моделі, тому ми можемо легко працювати з даними, що мають сотні полей (характеристик). Швидкість прогнозування значно більша, ніж швидкість тренування, оскільки є можливість зберігання розбиття дерев рішень для майбутнього використання.
12. Надійність до виключень і нелінійних даних. Random Forest обробляє виключення, по суті ігноруючи їх. Він також справляється з нелінійними характеристиками, що не несуть ніякої користі для прогнозування, тож мають бути проігноровані алгоритмом.
13. Обробляє незбалансовані дані. Алгоритм має методи балансування помилки в несиметричних наборах даних щодо класу. Random Forest намагається звести до мінімуму загальний коефіцієнт помилок, тому, коли ми маємо дисбалансований набір даних, більший (важливіший) клас отримає низький рівень помилок, а менший (менш важливий) - більше.
14. Низьке зміщення, Помірна дисперсія. Кожне дерево рішень має високу дисперсію, але низьке зміщення. Але оскільки ми усереднюємо всі дерева у Random Forest, ми усереднюємо дисперсію так, щоб мати модель з низьким зміщенням і помірною дисперсією.

#### Недоліки

1. Алгоритм схильний до перенавчання на деяких завданнях, особливо з великою кількістю шумів. Це може бути виправлено налаштуванням гіперпараметрів.

					ІА351.070БАК.001 ПЗ	Аркуш
						42
Зм	Арк.	№ документа	Підпис	Дата		

2. Великий розмір отримуваних моделей. Потрібно  $O(NK)$  пам'яті для зберігання моделі, де  $K$  — число дерев.
3. Модель інтерпретації: моделі Random Forest не є такими, що інтерпретуються; вони схожі на чорні ящики.

### Реалізації

1. Авторська реалізація Брейман і Катлер на мові Fortran 77
2. Пакет randomForest для R - портована версія оригінального авторського коду в R
3. Пакет party для R, містить модифікацію алгоритму
4. Існують реалізації алгоритму в системах Weka і RapidMiner
5. Реалізація модифікації алгоритму на [alglib.sources.ru](http://alglib.sources.ru)
6. FastRandomForest

## 3.2 Опис користувацької частини

Для користувачів Salesforce CRM було розроблене додаток, який вони можуть відкрити в App Launcher, де знаходяться усі доступні додатки системи (рис. 4.2.1):

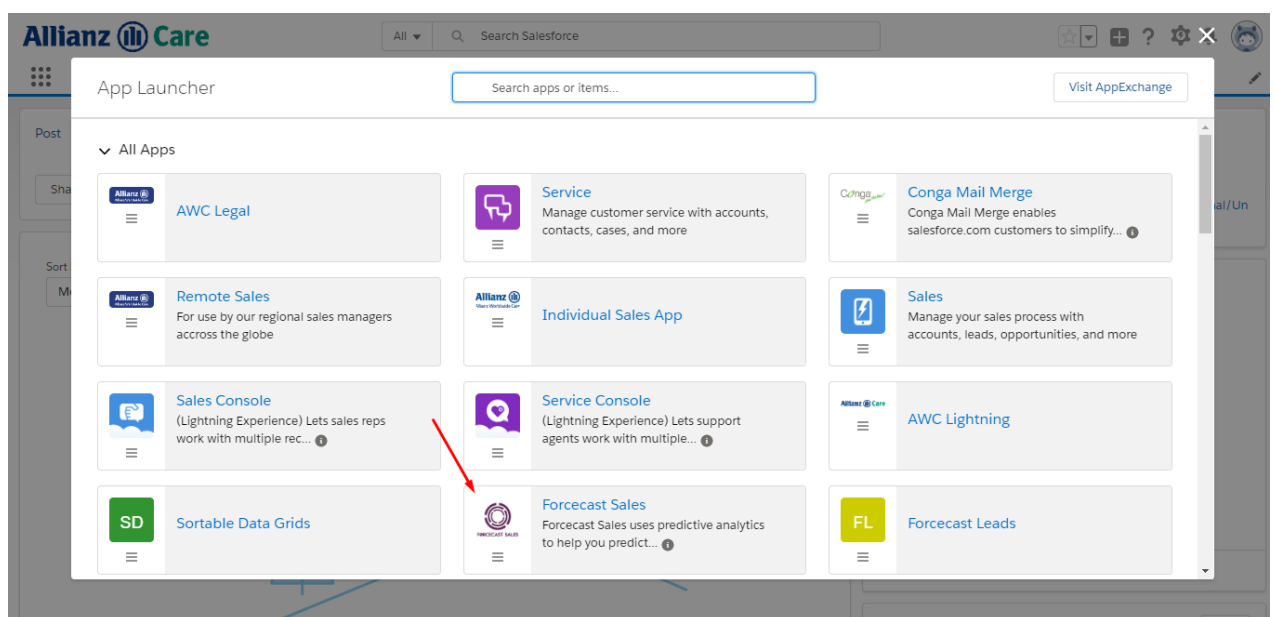


Рисунок 4.2.1 – Додаток з точки зору користувача Salesforce CRM

					<i>IA351.070BAK.001 ПЗ</i>	Аркуш
Зм	Арк.	№ документа	Підпис	Дата		43

Його можуть бачити тільки користувачі з певним рівнем доступу, який може надати адміністратор – тож їм користуються ті співробітники компанії, яким він потрібен (керівник може вирішувати, хто може чи не може користуватися додатком).

Додаток має наступні вкладки (рис. 4.2.2):

1. Домашня сторінка з основною інформацією щодо продаж за останній місяць/квартал/рік для усіх користувачів CRM, чи тільки даного співробітника та його підлеглих (з можливостями переключення).
2. Сторінка для налаштування додатку, де адміністратор може змінювати параметри аналізу, розпочати новий аналіз або налаштувати час, коли аналіз починається автоматично.
3. Сторінка перегляду усіх аналізів, що відбувалися останнім часом (рис. 4.2.4)
4. Сторінка перегляду усіх клієнтів (з даними результатів аналізів)
5. Сторінка перегляду усіх угод (з даними результатів аналізів та прогнозів щодо кожної конкретної угоди) (рис. 4.2.5)

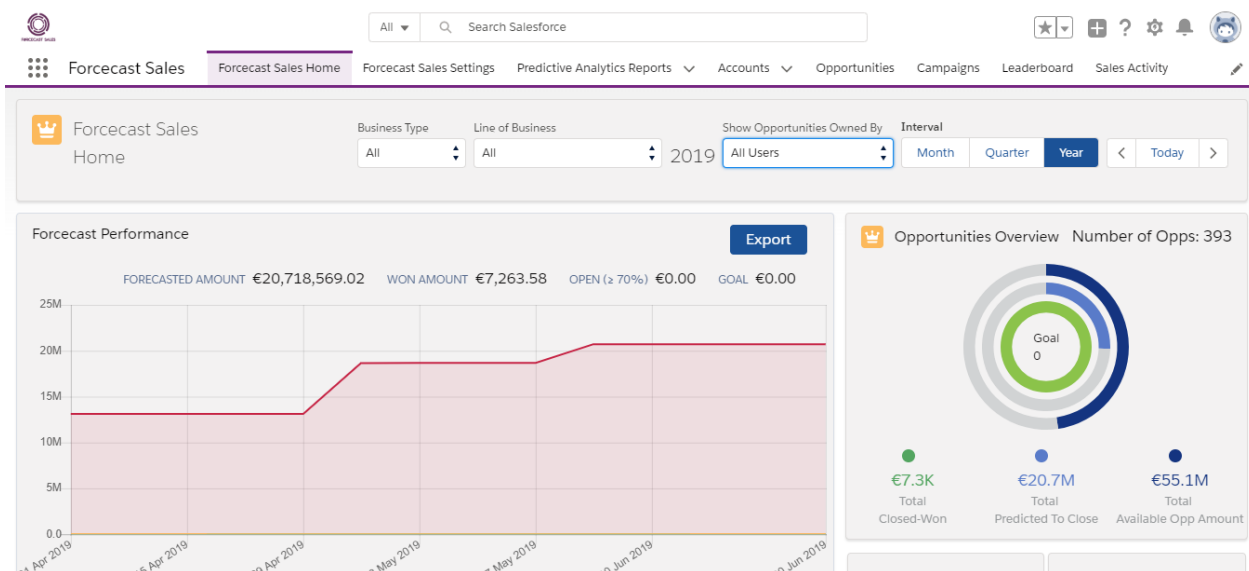


Рисунок 4.2.2 – Зовнішній вигляд додатку

6. Сторінка перегляду усіх комунікацій співробітника та клієнтів (з прогнозами та пропозиціями щодо дій, які потрібно зробити для кожного конкретного клієнта)

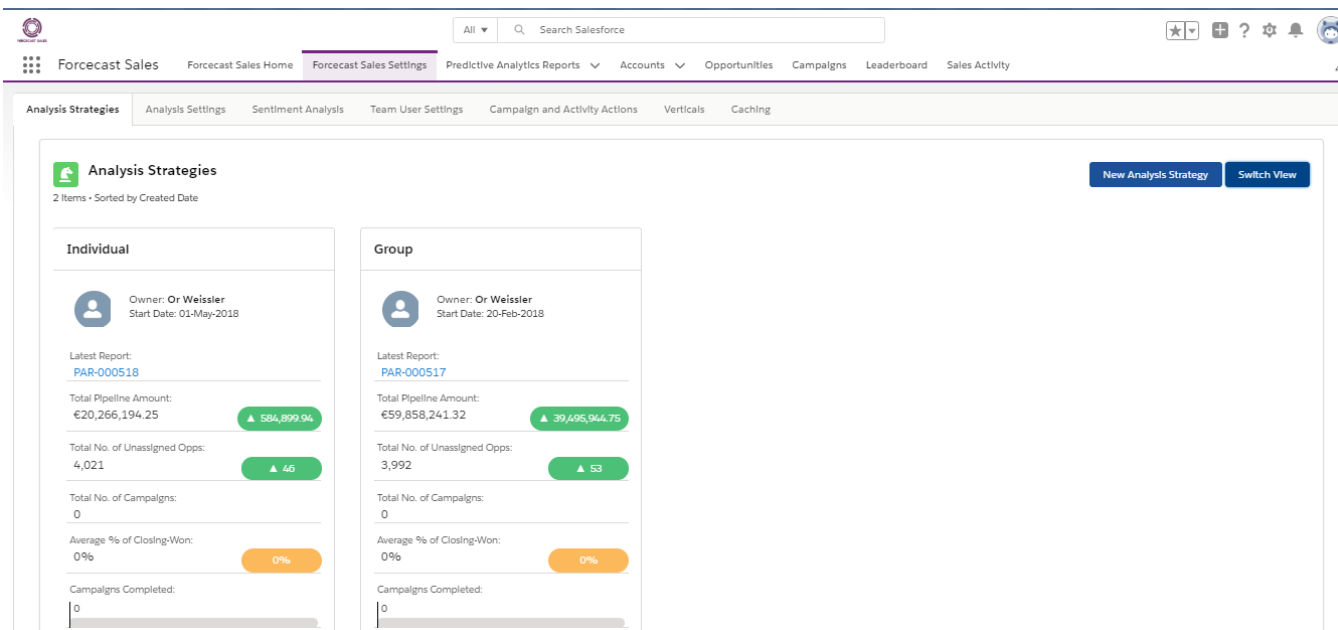


Рисунок 4.2.3 – Сторінка налаштувань додатку

7. Сторінка перегляду досягнень усієї команди
8. Сторінка перегляду власних досягнень даного співробітника.

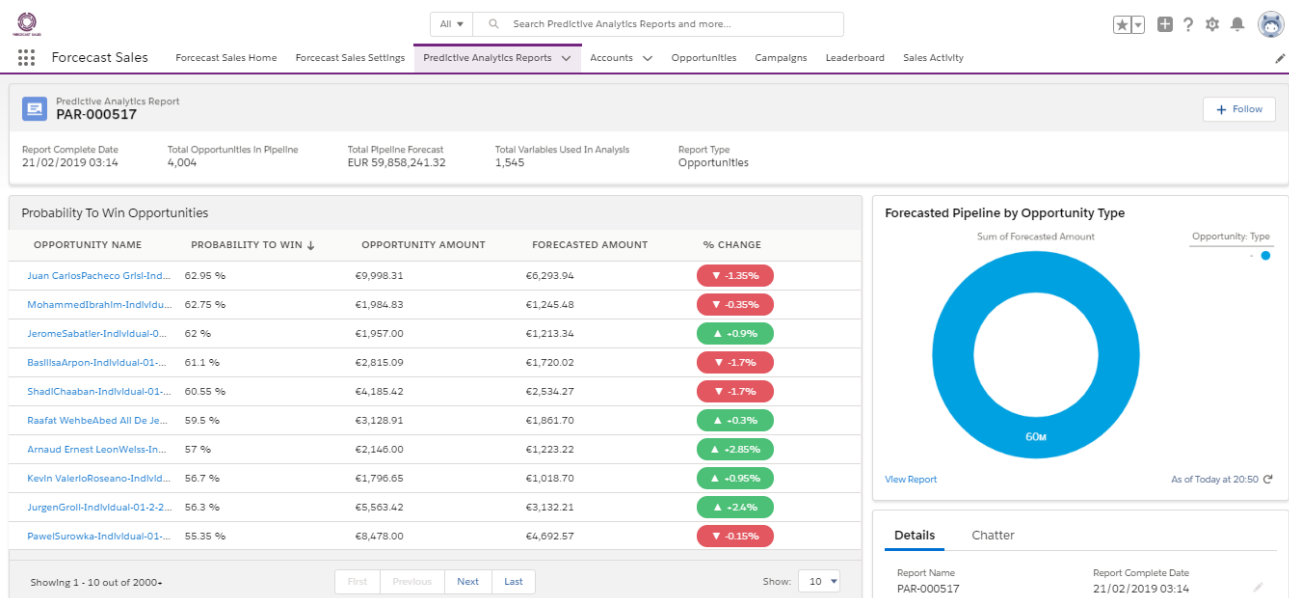


Рисунок 4.2.4 – Сторінка перегляду аналізу

У верхній частині сторінки можна побачити загальну інформацію щодо аналізу: дата та час, загальна кількість аналізованих угод, передбачений дохід, кількість змінних, що використані для аналізу та час, витрачений на аналіз.

Центральну частину сторінки займає список усіх аналізованих угод, прогнозовану вірогідність їх успіху, прогнозований дохід кожної угоди та зміна вірогідності успіху щодо попереднього аналізу (як змінюється вірогідність успіху угоди з часом).

Справа відображається графік розподілення прибутку від типу угоди.

Сторінка перегляду є здебільшого стандартною, в додаток було додано частину у правій колонці, яка відображає прогнозовану вірогідність успіху угоди, прогнозований дохід з угоди та головні причини, які впливають на вірогідність та важливість цих причин, яка була визначена алгоритмом.

На самій сторінці угоди зображено головну інформацію щодо угоди: посилання на сторінку клієнта, дохід, посилання на співробітника

Зм	Арк.	№ документа	Підпис	Дата

IA351.070БАК.001 ПЗ

Аркуш

46

компанії, що відповідає за цю угоду, її стадія та інша інформація, яка є індивідуальною для кожної окремої компанії.

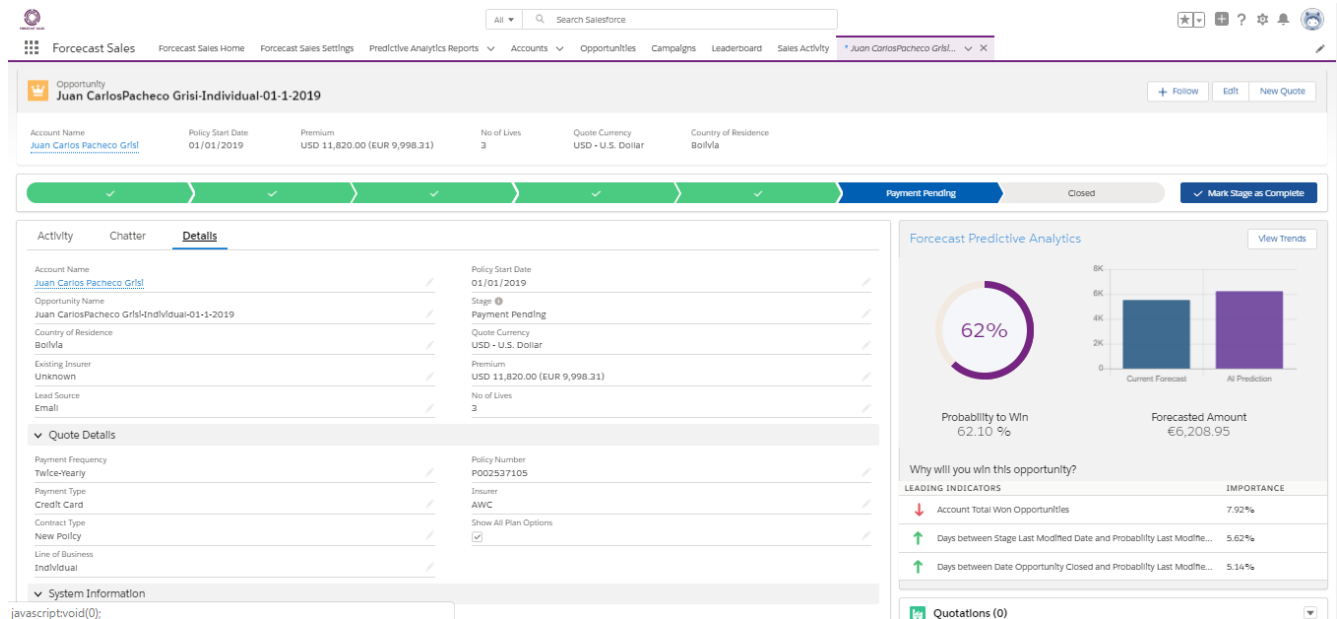


Рисунок 4.2.5 – Сторінка перегляду угоди

Отже, маємо реалізований користувацький інтерфейс, що задовольняє усім вимогам щодо можливостей співробітників компанії користуватися системою аналізу та обробки даних в повсякденному користуванні CRM системою.

### 3.3 Опис технології розробки

#### 3.3.1 Реалізація Random Forest мовою програмування R.

Мова програмування R має вбудовану бібліотеку randomForest, яку я використала для реалізації алгоритму.

Перш за все, завантажуюмо необхідні бібліотеки (використаємо тестовий набір даних для демонстрації алгоритму)

```
require(randomForest)
require(MASS) #Package which contains the Boston housing dataset
attach(Boston)
set.seed(101)

dim(Boston)
```

Наступним кроком відокремлюємо підмножину даних для навчання від підмножини даних для тестування

```
#training Sample with 300 observations
train=sample(1:nrow(Boston),300)
?Boston #to search on the dataset
```

Далі створюємо дерева та записуємо їх у змінну medv

```
Boston.rf=randomForest(medv ~ ., data = Boston , subset = train)
Boston.rf
##
## Call:
## randomForest(formula = medv ~ ., data = Boston, subset = train)
##      Type of random forest: regression
##      Number of trees: 500
## No. of variables tried at each split: 4
##
##      Mean of squared residuals: 12.62686
##      % Var explained: 84.74
```

Отже, приблизно 2/3 даних було використано для навчання, та 1/3, що залишилася, була використана для перевірки дерев (дані розподіляються випадковим чином). Також, кількість характеристик даних, випадковим чином вибраних на кожному кроці розбиття, є 4 ознаки.

Візуалізуємо граф, що репрезентує залежність кількості помилок, допущених деревами від кількості створених дерев (усього було створено 500 дерев):

```
plot(Boston.rf)
```

Зм	Арк.	№ документа	Підпис	Дата

ІА351.070БАК.001 ПЗ

Аркуш

48



Результат бачимо на рисунку (рис. 3.1.1):

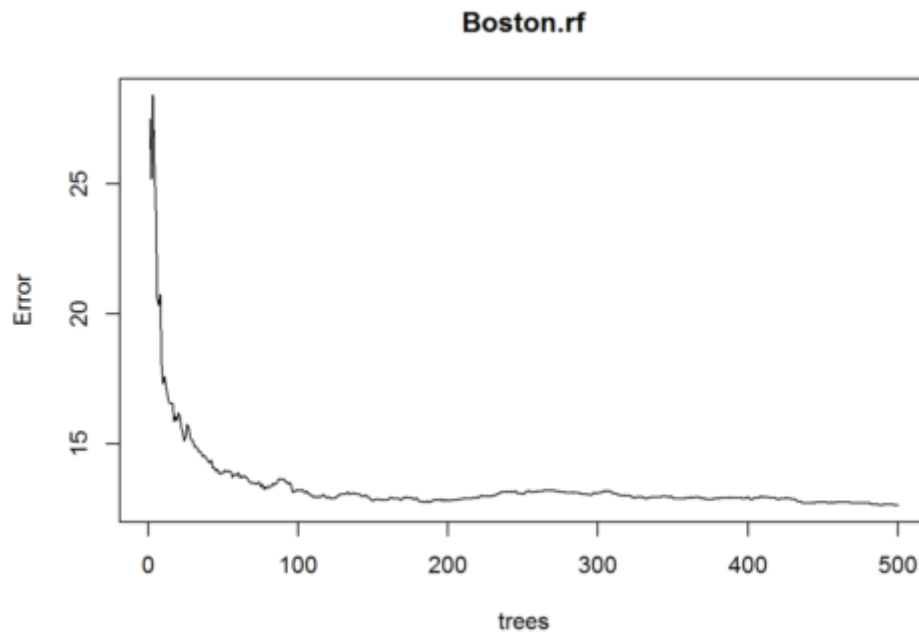


Рисунок 3.1.1 – Залежність кількості помилок від кількості дерев, створених алгоритмом Random Forest

Очевидно, що існує залежність між кількістю дерев та ефективністю моделі даних, чим більше дерев створено – тим краще працює прогнозування моделі.

Попередня модель вибирала випадковим чином тільки 4 характеристики для кожного розбиття. Тепер спробуємо використати усі 13 предикторів (стільки характеристик мають тестові дані), що можуть сформуватися на кожному розбитті:

```
oob.err=double(13)
test.err=double(13)

#mtry is no of Variables randomly chosen at each split
for(mtry in 1:13)
{
  rf=randomForest(medv ~ . , data = Boston , subset =
train,mtry=mtry,ntree=400)
```

```

oob.err[mtry] = rf$mse[400] #Error of all Trees fitted

pred<-predict(rf,Boston[-train,]) #Predictions on Test Set for each Tree
test.err[mtry]= with(Boston[-train,], mean( (medv - pred)^2)) #Mean
Squared Test Error

cat(mtry," ") #printing the output to the console

}

```

Бачимо відсоток помилок (в залежності від кількості побудованих дерев) для першої тестової моделі:

```
## [1] 26.06433 17.70018 16.51951 14.94621 14.51686 14.64315 14.60834
## [8] 15.12250 14.42441 14.53687 14.89362 14.86470 15.09553
```

Та для другої моделі (з застосування усіх 13 можливих предикторів) (out-of-bag error estimation):

```
## [1] 19.95114 13.34894 13.27162 12.44081 12.75080 12.96327 13.54794
## [8] 13.68273 14.16359 14.52294 14.41576 14.69038 14.72979
```

Тепер будуємо графік залежності кількості помилок від кількості «виращених» дерев для обох випадків на одній координатній площині:

```

matplot(1:mtry , cbind(oob.err,test.err), pch=19 ,
col=c("red","blue"),type="b",ylab="Mean Squared Error",xlab="Number of
Predictors Considered at each Split")
legend("topright",legend=c("Out of Bag Error","Test Error"),pch=19,
col=c("red","blue"))

```

Результат бачимо на рисунку (рис. 3.1.2):

Зм	Арк.	№ документа	Підпис	Дата

**IA351.070БАК.001 ПЗ**

Аркуш

50

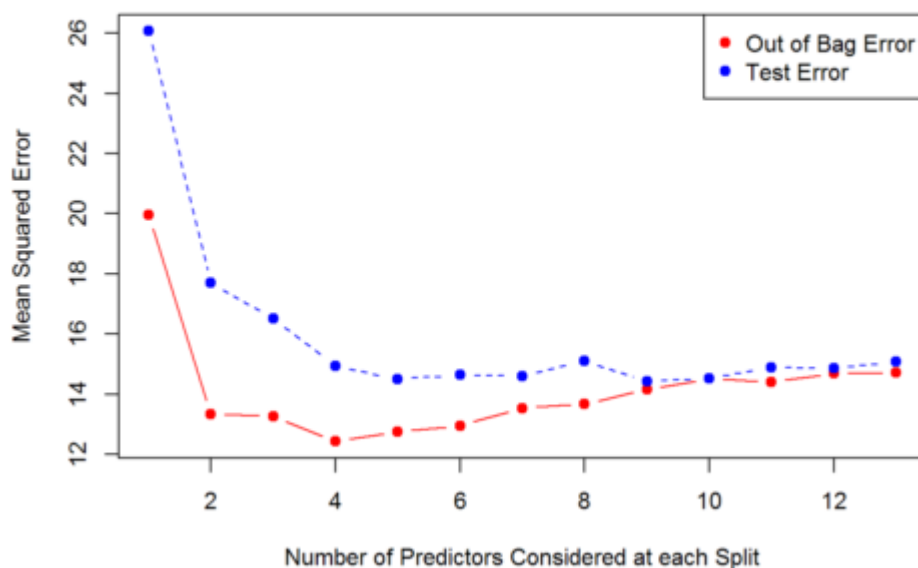


Рисунок 3.1.2 – Залежність кількості помилок від кількості дерев для різної кількості використаних предикторів

Червоним на графіку зображено кількість очікуваних помилок в залежності від кількості створених дерев для Out of Bag Error Estimates, блакитним – реальна кількість помилок, допущених алгоритмом на тестовій вибірці даних. Бачимо, що реальні результати відповідають очікуванню, та є задовільними.

В результаті, був реалізований такий складний та надійний алгоритм, як Random Forest з використанням корисної бібліотеки мовою програмування R – randomForest.

Лістинг програми, в якій реалізовано алгоритм Random Forest наведено у Додатку Ж.

### 3.3.2 Засоби інтеграції Salesforce CRM з алгоритмом Випадкових лісів.

Наступний крок після реалізації власне алгоритму буде реалізація зв'язку цього алгоритму на мові R з обраною CRM – Salesforce.

Зм	Арк.	№ документа	Підпис	Дата

IA351.070БАК.001 ПЗ

Аркуш

51

Найлегше це зробити за допомогою бібліотек на мові програмування Python.

Отже, перш за все потрібно авторизуватися с CRM, для цього використовуємо бібліотеку SalesforceUser на мові програмування Python:

```
from models import SalesforceUser

class Auth:
    def __init__(self, data, prefix=""):
        self.org_id = prefix + data['org_id']
        self.token = data['churndown_token']

    def authenticate(self):
        user = SalesforceUser.query.filter_by(org_id= self.org_id).first()
        if user is None:
            return None
        elif user.pexlify_token != self.token:
            return None
        else:
            return user
```

Та робимо звичайний HTTP запит на сервер, де знаходяться бази даних Salesforce CRM (адреса сервера нам відома, її передає в собі запит на обробку даних, що приходить з CRM):

```
import requests
import json
from config import app_config

class SalesforceRequest:
    def __init__(self, sandbox_url, code):
        self.sandbox_url = sandbox_url
        self.code = code

    def call(self):
        host = 'https://' + self.sandbox_url + '.salesforce.com'
        path = "/services/oauth2/token"
        redirect_uri = app_config.BASE_SERVER_URL +
```

Зм	Арк.	№ документа	Підпис	Дата

**IA351.070БАК.001 ПЗ**

Аркуш

52

```

'/salesforce_connection/create_new'
url = host + path
payload = {
    'grant_type': 'authorization_code',
    'code': self.code,
    'client_id': app_config.CLIENT_ID,
    'client_secret': app_config.CLIENT_SECRET,
    'redirect_uri': redirect_uri
}
r = requests.post(url, data= payload)
result = json.loads(r.text)
result['host'] = host
return result

def refresh_token(self):
    host = self.sandbox_url
    path = "/services/oauth2/token"
    url = host + path
    payload = {
        'grant_type': 'refresh_token',
        'client_id': app_config.CLIENT_ID,
        'client_secret': app_config.CLIENT_SECRET,
        'refresh_token': self.code
    }
    r = requests.post(url, data=payload)
    return json.loads(r.text)

```

Надалі використовуємо бібліотеки Salesforce, SalesforceRequest, SalesforceClient для завантаження даних з серверів CRM для подальшої обробки (лістинг наведено у додатках).

Отримані дані передаємо власне класу, що реалізує алгоритм Random Forest на мові програмування R. Для цього використовуємо бібліотеки DataFrame, pandas2ri, robjects.[15] Лістинг наведений у Додатку Ж.

### 3.3.3 Засоби реалізації користувацького інтерфейсу.

Користувацький інтерфейс додатку створений як частина Salesforce CRM за допомогою вбудованого фреймворку – Salesforce

					<i><b>IA351.070БАК.001 ПЗ</b></i>	Аркуш
						53
Зм	Арк.	№ документа	Підпис	Дата		

Lightning, а також з використанням HTML, CSS та JavaScript (для візуальної частини) та мови програмування Apex (для операцій з базою даних та зв'язком з алгоритмом Random Forest.

За допомогою можливостей Apex, система відправляє запит на сервер, де відбуваються усі розрахунки алгоритму. Це реалізовано як звичайний HTTP запит з Apex на сервер (адреса зберігається в налаштуваннях CRM). Лістинг наведено в додатку.

### 3.3 Висновки

В цьому розділі було реалізовано додаток до CRM-системи, призначений для аналізу даних існуючих угод та передбачення вірогідності успіху угод на базі Salesforce CRM. Було наведено структуру системи в цілому, місце додатку в системі, описано алгоритм роботи системи з додатком та алгоритм роботи додатку. Був розглянутий алгоритм аналізу і прогнозування Random Forest, технології реалізації цього алгоритму на мові програмування R. Був реалізований сучасний та інтуїтивно-зрозумілий користувацький інтерфейс, що надає можливість користувачу CRM-системи взаємодіяти з додатком.

## ВИСНОВКИ

В першому розділі було проаналізовано найбільш розповсюджені CRM-системи, на основі яких можливо розробити додаток, а саме Salesforce CRM, HubSpot CRM, Microsoft Dynamics CRM, Bitrix24, їх переваги, недоліки та функціональні можливості для реалізації додатку для аналізу та обробки даних. В результаті було вибрано Salesforce CRM через її популярність, унікальні можливості для створення персональних додатків та фреймворку, який вона надає розробникам програмного забезпечення для полегшення створення додатків.

В другому розділі було досліджено проблему прогнозування в сучасній підприємницькій діяльності, актуальність цієї проблеми була обґрунтована та формулюванні вимоги до продукту, що має вирішувати цю проблему.

В третьому розділі було реалізовано програмний додаток до вибраної CRM-системи Salesforce, що задовольняє поставленим вимогам, було розроблено і реалізовано користувацький інтерфейс, та алгоритм аналізу та прогнозування Random Forest, також зв'язок між ними.

Розроблений програмний продукт є адаптованим для роботи в умовах CRM-системи, володіє універсальністю, надійністю і стійкістю до зміни умов. Крім того, і це істотно, можна збільшити число вирішуваних завдань. Так, наприклад, при прогнозуванні обсягів продажів можна вирішити проблему впливу кожного показника (реклами, виставок, інтернету) на величину прибутку.

Було розроблено зручний та інтуїтивно-зрозумілий користувацький інтерфейс, за допомогою якого будь-який користувач CRM-системи може проводити такі операції з додатком, як налаштування параметрів аналізу, розпочати аналіз (або налаштувати час, коли цей аналіз розпочнеться автоматично), переглядання

					<i><b>IA351.070BAK.001 ПЗ</b></i>	Аркуш
						55
Зм	Арк.	№ документа	Підпис	Дата		

результати аналізу для усіх клієнтів на угод в межах системи, спостерігати динаміку зміни результатів аналізу протягом часу (за наявності серії аналізів).

Одна з переваг проекту - його дешевизна. Програмні продукти з схожим функціоналом можуть дорого коштувати підприємцям, тож далеко не всі компанії дозволяють собі придбати інструменти аналізу та обробки інформації та прогнозу.

					ІА351.070БАК.001 ПЗ	Аркуш
						56
Зм	Арк.	№ документа	Підпис	Дата		



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. С. Шовкопляс. Как повысить конкурентоспособность при помощи CRM // Office — 2005. — № 3-4, с. 12-18.
2. Л. Синило. Сложнее, чем кажется — внедрение CRM // Новый Маркетинг — 2006. — № 3, с. 87-95.
3. М. Кадыков. Битва за клиента: «Фронт» и «Тыл» // Отдел Маркетинга — 2006. — № 10, с. 32-33.
4. Лидовская О. П. Оценка эффективности маркетинга и рекламы. Готовые маркетинговые решения. — СПб.: Питер, 2008. — 141 с.
5. Taber, David. Salesforce.com. Secrets of Success: Best Practices for Growth and Profitability. — Prentice Hall, 2009. — 504 p. — ISBN 978-0137140763.
6. Benioff, Marc and Adler, Carlye. Behind the Cloud: The Untold Story of How Salesforce.com Went from Idea to Billion-Dollar Company-and Revolutionized an Industry. — Jossey-Bass, 2009. — 304 p. — ISBN 978-0470521168.
7. Roush, Wade. The Customer Is Sometimes Wrong (англ.) // Technology Review. — MIT, 2005, October. — ISSN 0040-1692.
8. Вертакова Ю.В. Упреждающее управление на основе новых информационных технологий. — Курск.: [б.и.], 2001.-151с.
9. Алгоритм Машинного Навчання Random Forest [Електронний ресурс] — режим доступу: — [https://uk.wikipedia.org/wiki/Random\\_forest](https://uk.wikipedia.org/wiki/Random_forest)
10. Алгоритм машинного навчання Random Forest [Електронний ресурс] — режим доступу: — <https://habr.com/ru/post/320726/>
11. Айвазян С. А., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: основы моделирования и первичная обработка данных. — М.: Финансы и статистика, 1983.

					<i>IA351.070БАК.001 ПЗ</i>	Аркуш
						57
Зм	Арк.	№ документа	Підпис	Дата		

12. Айвазян С. А., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: исследование зависимостей. — М.: Финансы и статистика, 1985.
13. Айвазян С. А., Бухштабер В. М., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: классификация и снижение размерности. — М.: Финансы и статистика, 1989.
14. Вапник В. Н. Восстановление зависимостей по эмпирическим данным. — М.: Наука, 1979.
15. Журавлев Ю. И., Рязанов В. В., Сенько О. В. «Распознавание». Математические методы. Программная система. Практические применения. — М.: Фазис, 2006. ISBN 5-7036-0108-8.
16. Загоруйко Н. Г. Прикладные методы анализа данных и знаний. — Новосибирск: ИМ СО РАН, 1999. ISBN 5-86134-060-9.
17. Флах П. Машинное обучение. — М.: ДМК Пресс, 2015. — 400 с. — ISBN 978-5-97060-273-7.
18. Шлезингер М., Главач В. Десять лекций по статистическому и структурному распознаванию. — Киев: Наукова думка, 2004. ISBN 966-00-0341-2.
19. Hastie, T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. — 2nd ed. — Springer-Verlag, 2009. — 746 p. — ISBN 978-0-387-84857-0..
20. Mitchell T. Machine Learning. — McGraw-Hill Science/Engineering/Math, 1997. ISBN 0-07-042807-7.
21. Ryszard S. Michalski, Jaime G. Carbonell, Tom M. Mitchell (1983), Machine Learning: An Artificial Intelligence Approach, Tioga Publishing Company, ISBN 0-935382-05-4 (Machine Learning: An Artificial Intelligence Approach в «Книгах Google»).
22. Vapnik V. N. Statistical learning theory. — N.Y.: John Wiley & Sons, Inc., 1998. [1]

					<i>IA351.070БАК.001 ПЗ</i>	Аркуш
						58
Зм	Арк.	№ документа	Підпис	Дата		

23. Bernhard Schölkopf, Alexander J. Smola Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond. — MIT Press, Cambridge, MA, 2002 ISBN 978-0-262-19475-4 [2]
24. I. H. Witten, E. Frank Data Mining: Practical Machine Learning Tools and Techniques (Second Edition). — Morgan Kaufmann, 2005 ISBN 0-12-088407-0 [3]
25. Liang Wang, Li Cheng, Guoying Zhao. Machine Learning for Human Motion Analysis. — IGI Global, 2009. — 318 p. — ISBN 978-1-60566-900-7.
26. Алгоритми машинного навчання [Електронний ресурс] – режим доступу: – <https://appttractor.ru/info/articles/desyat-algoritmov-mashinnogo-obucheniya-kotoryie-vam-nuzhno-znat.html>
27. EverString Reviews: Overviews, Pricing and Features [Електронний ресурс] – режим доступу: <https://reviews.financesonline.com/p/everstring>
28. HubSpot – Internet Marketing Company – [Електронний ресурс] – режим доступу: <https://www.hubspot.com/our-story>
29. Microsoft PowerPlatform [Електронний ресурс] – режим доступу: <https://powerplatform.microsoft.com/en-us/>
30. Бітрікс 24. Що таке Бітрікс 24 [Електронний ресурс] – Режим доступу: <https://www.bitrix24.ua/whatisthis/>

ДОДАТОК А

Діаграма структури системи



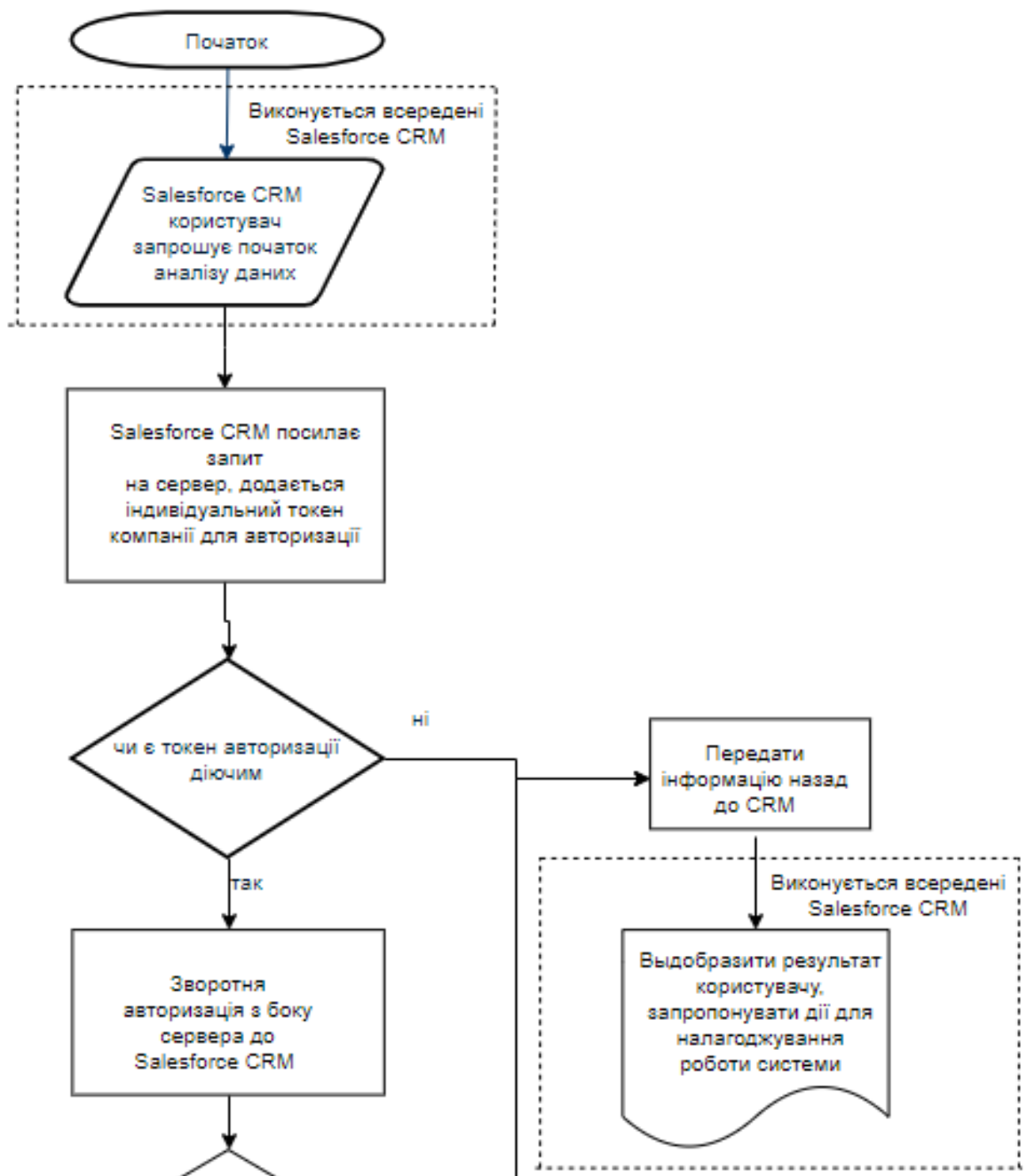
## ДОДАТОК Б

### Блок-схема алгоритму роботи CRM системи з додатком



## ДОДАТОК В

### Блок схема алгоритму роботи додатку

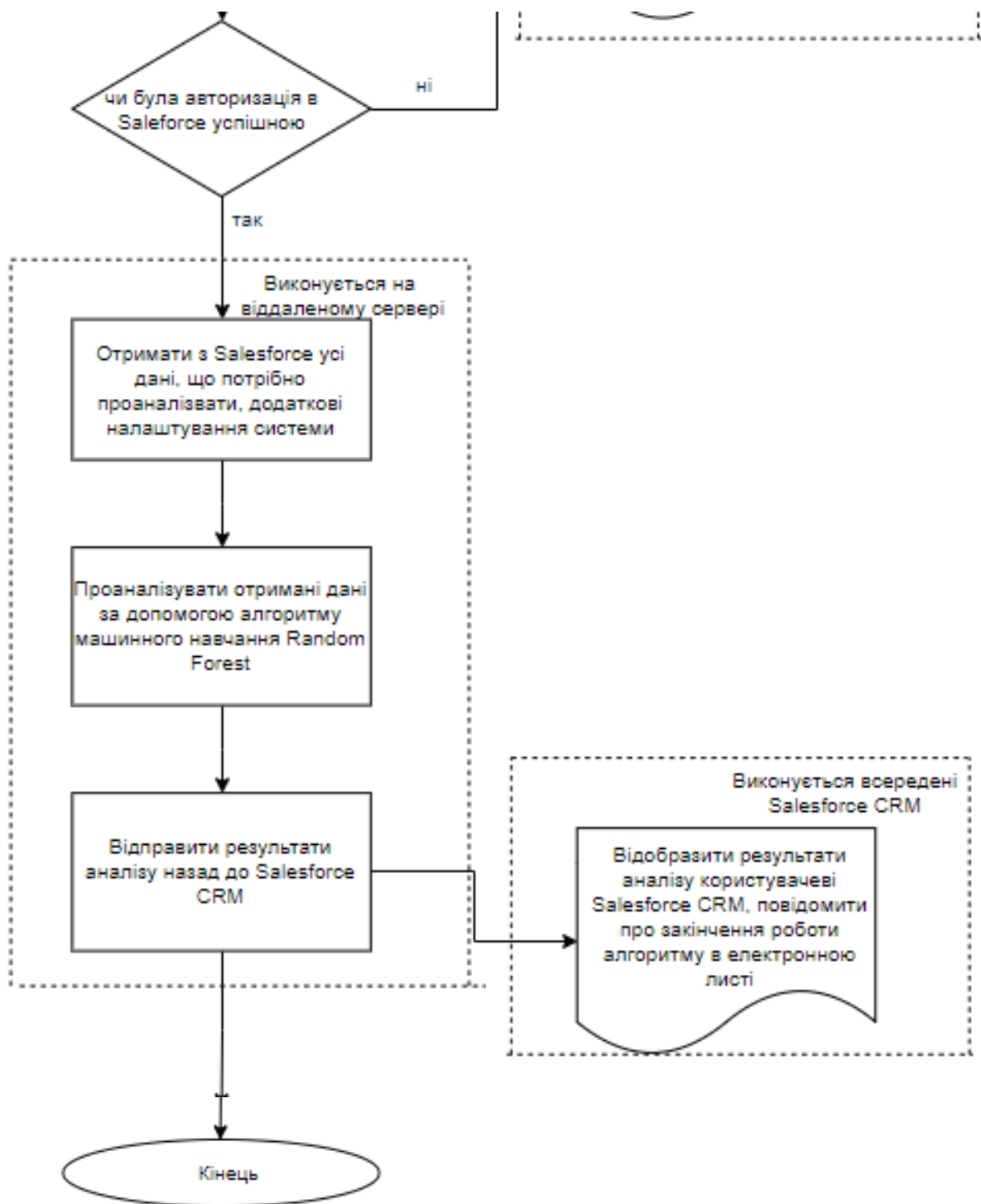


Зм	Арк.	№ документа	Підпис	Дата

ІА351.070БАК.001 ПЗ

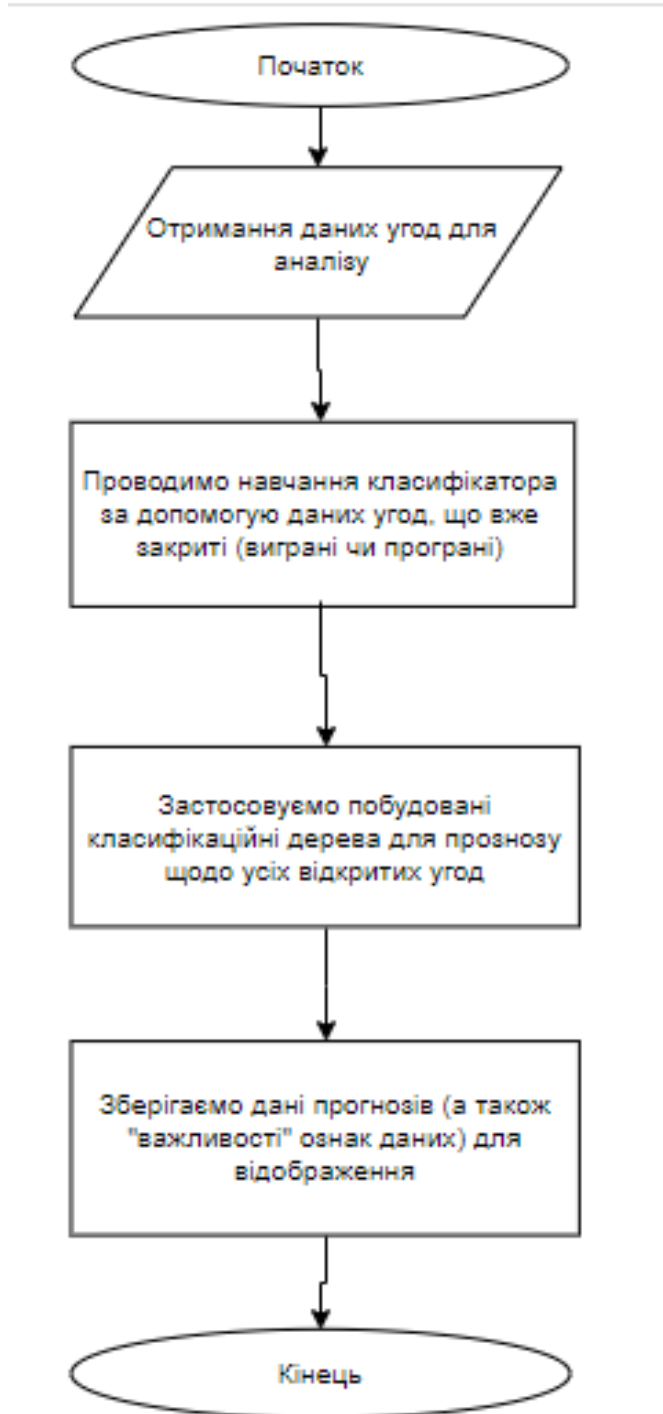
Аркуш

62



## ДОДАТОК Г

### Блок схема алгоритму Random Forest



Зм	Арк.	№ документа	Підпис	Дата

ІА351.070БАК.001 ПЗ

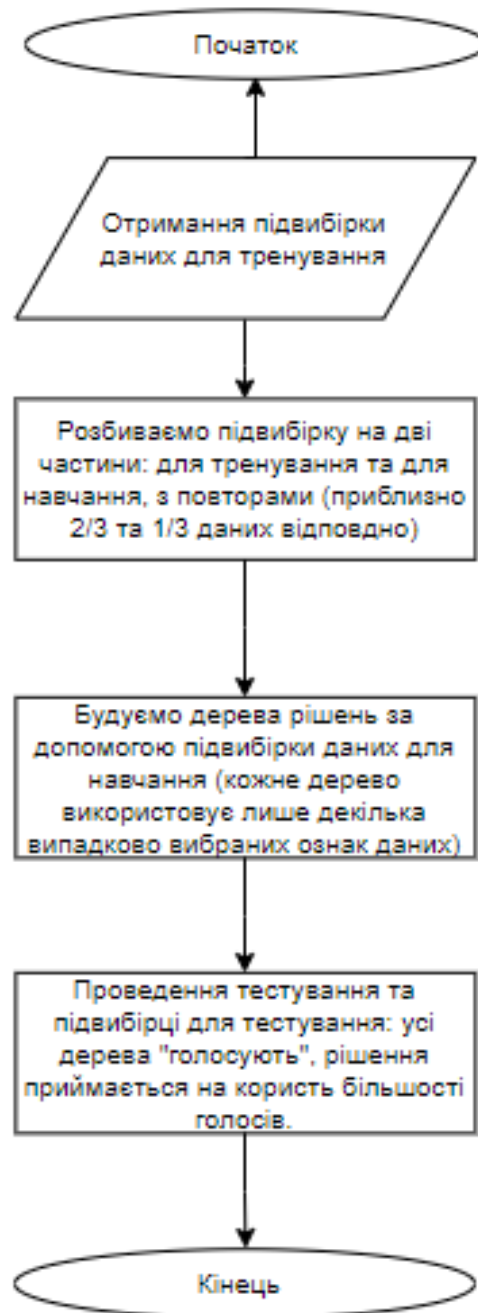
Аркуш

64



## ДОДАТОК Д

### Алгоритм навчання класифікатора Random Forest



Зм	Арк.	№ документа	Підпис	Дата

ІА351.070БАК.001 ПЗ

Аркуш

65

## ДОДАТОК Е

### Лістинг програми авторизації в системі Salesforce CRM

```
import requests
import json
from config import app_config

class SalesforceRequest:
    def __init__(self, sandbox_url, code):
        self.sandbox_url = sandbox_url
        self.code = code

    def call(self):
        host = 'https://' + self.sandbox_url + '.salesforce.com'
        path = "/services/oauth2/token"
        redirect_uri = app_config.BASE_SERVER_URL +
        '/salesforce_connection/create_new'
        url = host + path
        payload = {
            'grant_type': 'authorization_code',
            'code': self.code,
            'client_id': app_config.CLIENT_ID,
            'client_secret': app_config.CLIENT_SECRET,
            'redirect_uri': redirect_uri
        }
        r = requests.post(url, data= payload)
        result = json.loads(r.text)
        result['host'] = host
        return result

    def refresh_token(self):
        host = self.sandbox_url
        path = "/services/oauth2/token"
        url = host + path
        payload = {
            'grant_type': 'refresh_token',
            'client_id': app_config.CLIENT_ID,
            'client_secret': app_config.CLIENT_SECRET,
            'refresh_token': self.code
        }
        r = requests.post(url, data=payload)
        return json.loads(r.text)
```

					ІА351.070БАК.001 ПЗ	Аркуш
						66
Зм	Арк.	№ документа	Підпис	Дата		

## ДОДАТОК Ж

### Лістинг програми аналізу даних

```
from numpy.linalg import norm
import rpy2.robjects as robjects
from rpy2.robjects import pandas2ri
from pandas import DataFrame

class RandomForestController:
    def __init__(self, random_forest_formula):
        self.random_forest_formula = random_forest_formula
        pandas2ri.activate()

    def run(self, train_data, test_data, split_field):
        robjects.r("""
            f <- function(train_data, test_data) {
                library(randomForest)
                set.seed(12345)
                random_forest_formula <- "" + self.random_forest_formula + ""
                # force levels by joining the data together
                data <- rbind(train_data, test_data)

                # convert resulting characters to factors
                for(i in 1:ncol(data)) {
                    if(is.character(data[,i])) {
                        data[,i] <- factor(data[,i])
                    }
                }
                # split data into test and train again
                train_data <- data[1:nrow(train_data),]
                test_data <- data[(nrow(train_data)+1):nrow(data),]

                # remove variables with too many levels (more than 53 is too many)
                bad <- which(sapply(train_data, nlevels) > 53)
                if(length(bad) > 0) {
                    print(paste("removed features: ", paste(colnames(train_data)[bad],
collapse=", ")))
                    train_data <- train_data[,-bad]
                    test_data <- test_data[,-bad]
                }
                fit <- randomForest(as.formula(random_forest_formula), data=train_data,
importance=TRUE, ntree=2000, na.action=na.roughfix)
                Prediction <- predict(fit, test_data, type="vote", na.action=na.roughfix)
                imp <- importance(fit, class = NULL, scale = TRUE)
                train_data_split <- split(train_data, train_data$"" + split_field + "")
                train_data_won <- train_data_split[['1']]
                train_data_lost <- train_data_split[['0']]
            }
        """)
```

					ІА351.070БАК.001 ПЗ	Аркуш
						67
Зм	Арк.	№ документа	Підпис	Дата		

```

stats_name <- list()
stats_won <- list()
stats_lost <- list()

for(i in 1:ncol(train_data)) {
  if(is.factor(train_data[,i])) {
    varN <- toString(colnames(train_data)[i])
    lostA <- toString(names(which.max(table(train_data_lost[,i]))))
    wonA <- toString(names(which.max(table(train_data_won[,i]))))

    stats_name[[i]] <- varN
    stats_lost[[i]] <- lostA
    stats_won[[i]] <- wonA
  } else {
    varN <- toString(colnames(train_data)[i])
    lostA <- toString(mean(train_data_lost[,i]))
    wonA <- toString(mean(train_data_won[,i]))

    stats_name[[i]] <- varN
    stats_lost[[i]] <- lostA
    stats_won[[i]] <- wonA
  }
}
return(list(Prediction, imp, paste(stats_name, collapse = '--__--'),
paste(stats_lost, collapse = '--__--'), paste(stats_won, collapse = '--__--')))
})

rtrain = self.to_r(train_data)
rtest = self.to_r(test_data)

r_f = robjects.globalenv['f']
result = r_f(rtrain, rtest)
importance_data = pandas2ri.rizpy(result[1])[0]
result = {
  'odds_to_win': pandas2ri.rizpy(result[0])[0].tolist(),
  'importance_variables': (importance_data / norm(importance_data)).tolist(),
  'var_stats_name': pandas2ri.rizpy(result[2])[0].split('--__--'),
  'var_stats_lost': pandas2ri.rizpy(result[3])[0].split('--__--'),
  'var_stats_won': pandas2ri.rizpy(result[4])[0].split('--__--')
}
return result

def to_r(self, data):
  columns = data.pop(0)
  df = DataFrame.from_records(data, columns= columns)
  return pandas2ri.py2ri(df)

```

## ДОДАТОК 3

### Лістинг програми контролеру передачі даних між CRM та алгоритмом аналізу

```
from lib.salesforce_package import SalesforcePackage
from lib.random_forest_sk_controller import RandomForestSkController
from lib.util import list_uniq

import math

class OppReport(SalesForcePackage):

    def __init__(self, salesforce_user, report_id):
        SalesforcePackage.__init__(self, salesforce_user, "Opportunity", ["Id"], report_id)

    def get_salesforce_data_queries(self, fields_for_nested_objects, fields_for_opp):
        where_clause = "WHERE Id IN (SELECT " + self.prefix + "Opportunity__c FROM " +
        self.prefix + "Opportunity_Analysis_Data__c WHERE " + self.prefix +
        "Selected_For_Analysis__c = true)"
        return self.get_salesforce_queries(fields_for_nested_objects, fields_for_opp,
        where_clause, "Opportunity")

    def data_from_source(self):
        self.update_report_progress(self.report_id, 4)
        [fields_for_opp, fields_for_nested_objects, fields_for_analysis, datetime_fields,
        diff_datetime_fields] = self.fields()
        soql_queries = self.get_salesforce_data_queries(fields_for_nested_objects,
        fields_for_opp)
        self.update_report_progress(self.report_id, 6)
        return self.query_data_and_create_matrix(soql_queries, fields_for_analysis, 6, 64,
        "Opportunity", datetime_fields, diff_datetime_fields)

    def get_headers_of_all_parameters_to_run_random_forest_on(self, train_data,
        test_data):
        class FileData:
            def __init__(self, file_data):
                self.file_data = file_data[1:len(file_data)]
                self.columns = file_data[0]

            def is_column_data_all_numbers(self, column_index):
                for data in self.file_data:
                    if isinstance(data[column_index], (bool)) or not
                    isinstance(data[column_index], (int, float)):
                        return False
                return True
```

					ІА351.070БАК.001 ПЗ	Аркуш
						69
Зм	Арк.	№ документа	Підпис	Дата		

```

def is_column_data_all_same(self, column_index):
    last_data = None
    for data in self.file_data:
        if last_data is None:
            last_data = str(data[column_index])
        if str(data[column_index]) != last_data:
            return False
    return True

def number_of_different_options_in_column(self, column_index):
    different_options = list_uniq(list(map(lambda x: x[column_index],
self.file_data)))
    return len(different_options)

def all_columns(self):
    return self.columns

def __get_closed_stages():
    stage_header_index = -1
    for i, data_header in enumerate(train_data[0]):
        if data_header.lower() == 'stagename':
            stage_header_index = i
    closed_stages = []
    for value in train_data:
        if not value[stage_header_index] in closed_stages:
            closed_stages.append(value[stage_header_index])
    return closed_stages

file_data = FileData(train_data + test_data[1:len(test_data)])
closed_stages = __get_closed_stages()
fields_to_exclude_for_analysis =
self.get_fields_to_exclude_for_analysis(closed_stages)
all_columns_headers = file_data.all_columns()
all_headers_of_parameters = []
fields_to_exclude_for_analysis = fields_to_exclude_for_analysis + ["id", "iswon",
"isclosed", "stagename",
                                "forecastcategory", "forecastcategoryname",
                                "forecastcategory", "forecastcategoryname"]

column_index = 0
while column_index < len(all_columns_headers):
    if (not (all_columns_headers[column_index].lower() in
fields_to_exclude_for_analysis)) and
(file_data.is_column_data_all_numbers(column_index) or
file_data.number_of_different_options_in_column(column_index) < 53) and not
file_data.is_column_data_all_same(column_index):
        if not all_columns_headers[column_index] is None:
            all_headers_of_parameters.append(all_columns_headers[column_index])

```

```

        column_index += 1
    return all_headers_of_parameters

def get_column_name_index(self, file_data, column_name):
    all_columns_headers = file_data[0]
    column_index = 0
    customer_column_name_index = -1
    while column_index < len(all_columns_headers):
        if all_columns_headers[column_index] == column_name:
            customer_column_name_index = column_index
            column_index = column_index + 1
    return customer_column_name_index

def get_opp_column_id_index(self, file_data):
    return self.get_column_name_index(file_data, "Id")

def get_random_forest_formula(self, all_headers_of_parameters, learn_variable):
    churn_field = "as.factor(" + learn_variable + ")"
    random_forest_formula = churn_field + "~ " + " +
".join(all_headers_of_parameters)
    return random_forest_formula

def save_probabilities_under_the_report(self, odds_to_win, opp_column_id_index,
file_data):
    i = 0
    progress_step = 10.0 / len(file_data)
    progress = 80.0
    last_progress = 80.0
    opp_results = []
    while i < len(odds_to_win):
        opp_id = file_data[i][opp_column_id_index]
        if odds_to_win[i] is None:
            probability_to_win = 0
        else:
            probability_to_win = odds_to_win[i]
        opp_result = {
            self.prefix + "Opportunity__c" : opp_id,
            self.prefix + "Churn_Report__c" : self.report_id,
            self.prefix + "Probability_to_Win__c" : probability_to_win }
        opp_results.append(opp_result)
        i = i+1
        progress += progress_step
        progress_update = int(round(progress))
        if progress_update > last_progress:
            self.update_report_progress(self.report_id, progress_update)
            last_progress = progress_update
    self.bulk_create(self.prefix + "Opportunity_Analyse__c", opp_results)

```

Зм	Арк.	№ документа	Підпис	Дата

**IA351.070БАК.001 ПЗ**

Аркуш

71

```

def save_importance_of_parameters_under_the_report(self,
all_headers_of_parameters, split_field, prediction_data):

    hash_between_field_name_and_value =
self.get_hash_between_field_name_and_value("Opportunity")
    importance_variables = prediction_data['importance_variables']
    var_stats_name = prediction_data['var_stats_name']
    var_stats_lost = prediction_data['var_stats_lost']
    var_stats_won = prediction_data['var_stats_won']
    var_stats = {}
    for i in range(0, len(var_stats_name)):
        var_stats[var_stats_name[i].lower()] = [var_stats_lost[i], var_stats_won[i]]
    all_variables = []
    i = 0
    progress_step = 10.0 / len(importance_variables)
    progress = 90.0
    last_progress = 90.0
    number_sf_field_types = ["currency", "int", "double", "percent"]
    while i < len(importance_variables):
        field_name = all_headers_of_parameters[i].lower()
        if (field_name in hash_between_field_name_and_value and field_name !=
'probability'):
            variable_name = str(hash_between_field_name_and_value[field_name][0])
            variable_type = hash_between_field_name_and_value[field_name][1]
            [lost_avg, won_avg] = var_stats[field_name]
            if i > len(importance_variables) or importance_variables[i] is None or
math.isnan(importance_variables[i]):
                importance_variable = 0
            else:
                importance_variable = importance_variables[i]
            if variable_type in number_sf_field_types:
                new_variable = {
                    self.prefix + "Won_Avg_Value_Number__c" : float(won_avg),
                    self.prefix + "Lost_Avg_Value_Number__c" : float(lost_avg),
                    self.prefix + "Variable_Api_Name__c" : field_name,
                    "Name" : variable_name,
                    self.prefix + "Churn_Report__c" : self.report_id,
                    self.prefix + "Importance__c" : importance_variable
                }
                all_variables.append(new_variable)
            else:
                if variable_type != "boolean":
                    if won_avg == "0":
                        won_avg = "Blank"
                    if lost_avg == "0":
                        lost_avg = "Blank"
                new_variable = {
                    self.prefix + "Won_Avg_Value_Text__c" : str(won_avg),

```

					ІА351.070БАК.001 ПЗ	Аркуш
						72
Зм	Арк.	№ документа	Підпис	Дата		



```

        self.prefix + "Lost_Avg_Value_Text__c" : str(lost_avg),
        self.prefix + "Variable_Api_Name__c" : field_name,
        "Name" : variable_name,
        self.prefix + "Churn_Report__c" : self.report_id,
        self.prefix + "Importance__c" : importance_variable
    }
    all_variables.append(new_variable)
    i = i+1
    progress += progress_step
    progress_update = int(round(progress))
    if progress_update > last_progress:
        self.update_report_progress(self.report_id, progress_update)
        last_progress = progress_update
    self.bulk_create(self.prefix + "Churn_Report_Variable__c", all_variables)

def run(self):
    [train_data, test_data] = self.data_from_source()
    all_headers_of_parameters =
self.get_headers_of_all_parameters_to_run_random_forest_on(train_data,
test_data)
    if len(all_headers_of_parameters) == 0:
        self.update_report_error(self.report_id, 'You choose to exclude all of the
variables for the analysis', 'Known')
        return False
    opp_column_id_index = self.get_opp_column_id_index(train_data)
    random_forest_formula =
self.get_random_forest_formula(all_headers_of_parameters, "iswon")
    forest = RandomForestSkController(random_forest_formula)
    prediction_data = forest.run(train_data, test_data)
    odds_to_win = prediction_data['odds_to_win']
    self.save_probabilities_under_the_report(odds_to_win, opp_column_id_index,
test_data)

self.save_importance_of_parameters_under_the_report(all_headers_of_parameters,
"iswon", prediction_data)
    self.update_report_done(self.report_id)

def log_exception(self, e):
    self.update_report_error(self.report_id, repr(e), 'Unknown')

```